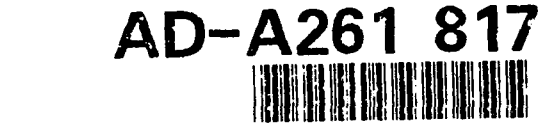


(2)

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

DEVELOPMENT OF RADAR ALGORITHMS
FOR INSTRUCTIONAL USE AT USNPGS

by

Paul A. Ohrt

December 1992

Thesis Advisor: Gurnam S. Gill

Approved for public release; distribution is unlimited

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b OFFICE SYMBOL (If applicable) EC	7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000	7b ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000
8a NAME OF FUNDING/SPONSORING ORGANIZATION 8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c ADDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO / PROJECT NO / TASK NO / WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification)
DEVELOPMENT OF RADAR ALGORITHMS FOR INSTRUCTIONAL USE AT USNPGS

12 PERSONAL AUTHOR(S)
OHRT, Paul A.

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Master's Thesis	FROM _____ TO _____	December 1992	128

16 SUPPLEMENTARY NOTATION
The views expressed in thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the US Government

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	simulation; DFT; window; pulse compression; doppler

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

This thesis is concerned with the use of simulation in the teaching of radar signal processing (RSP). The aspects of RSP to be investigated and simulated are the development of the DFT as a filter bank for radar applications, filter sidelobe reduction by use of weighted DFT doppler filter banks and the generation of pulse compression coding schemes under doppler conditions. Simulation programs have been written in MATLAB for the above processes. These simulations are cost effective, convenient to use and easy to reproduce since they are run on personal computers that are readily available to students. Computer simulation provides a learning environment which cannot be achieved by traditional methods alone and complements classroom teaching. In particular, it aids the student by removing a major computational burden which allows the student to explore the non-trivial real world problems which he could not do before.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT	21 ABSTRACT SECURITY CLASSIFICATION	
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	UNCLASSIFIED	
22a NAME OF RESPONSIBLE INDIVIDUAL GILL, Gurnam S.	22b TELEPHONE (Include Area Code) 408-646-2652	22c OFFICE SYMBOL EC/Gl

DD Form 1473, JUN 86 *Previous editions are obsolete*

Approved for public release; distribution is unlimited.

Development of Radar Algorithms
for Instructional Use at the
United States Naval Post Graduate School

by

Paul A. Ohrt
Major, Canadian Army
B.S., Royal Military College of Canada

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1992

Author: Paul A Ohrt
Paul A. Ohrt

Approved by: L. S. Gill
G.S. Gill, Thesis Advisor

M. Tummala
M. Tummala, Second Reader

Michael A. Morgan
Michael A. Morgan, Chairman
Department of Electrical and Computer Engineering

ABSTRACT

This thesis is concerned with the use of simulation in the teaching of radar signal processing (RSP). The aspects of RSP to be investigated and simulated are the development of the DFT as a filter bank for radar applications, filter sidelobe reduction by the use of weighted DFT doppler filter banks and the generation of pulse compression coding schemes under doppler conditions. Simulation programs have been written in MATLAB for the above processes. These simulations are cost effective, convenient to use and easy to reproduce since they are run on personal computers that are readily available to students. This provides a learning environment which can not be achieved by traditional methods alone and compliments classroom teaching. In particular it aids the student by removing a major computational burden which allows the student to explore the non-trivial real world problems which he could not do before.

TABLE OF CONTENTS

I. INTRODUCTION

A. OVERVIEW OF THESIS

This thesis will concentrate on three areas of radar signal processing (RSP) and how they can be simulated to aid student understanding. The primary research done was to develop algorithms for the display and manipulation of several radar signal processes in order to provide students taking radar courses a fuller understanding of the processes involved. Aspects of radar signal processing that were investigated were the development of the DFT as a filter bank for radar situations, the use of windows to create weighted DFT filter banks and the generation of pulse compression coding schemes under doppler conditions.

1. The DFT as a Filter Bank

The background as to how the DFT is used to form a filter bank from an input sinusoid is examined. The input sinusoids will be either a complex sinusoid from quadrature sampling or a real sinusoid from single channel sampling. The reason only sinusoids are treated is because it is a sinewave carrier that needs to be detected and processed in most radar situations. The student will be able to select the pulse repetition frequency (PRF) at which the signal is to be sampled and the number of DFT filters that are to be

considered (N) besides various other parameters. Outputs will include a graphical display of a single DFT filter response for a range of sinusoidal inputs, an overall DFT filter bank response to a range of sinusoidal inputs and the output of the DFT filter bank if a single target were to appear at a specified doppler frequency.

2. Weighted Doppler Filter Bank

In order to reduce the effects of the sidelobes of the filters, the input sequence to the DFT is weighted in a various ways that are called windows. The student will be able to select the windowing function he wishes to view for various input lengths and PRFs. The outputs will be displayed graphically versus the output of a rectangular window for comparison purposes.

3. Simulation of Pulse Compression Waveforms

There will be a brief discussion on the various pulse compression modulation schemes available and their characteristics. Then the simulation program will be used to demonstrate how some codes are doppler tolerant. Codes to be generated are the Barker, combined Barker, pseudo random and two of the various polyphase schemes (the P3 code and the Frank). The student will be able to select the code he wishes to observe, the conditions under which he wishes to observe it and what doppler effects (if any) are to be allowed. Outputs will include a graphical display of the

compressed waveform and the main parameters associated with the pulse compression. In addition a subprogram to show the effects of eclipsing loss is available.

4. Reasons For Simulation

Older analog radars usually involved no more than a transmitter and a receiver connected to a video display. This simple and direct system was easy to teach as signal processing was at a minimum. However, modern radars are much more complex and employ sophisticated radar signal processing techniques to extract weak target signals from large interference due to rain or ground clutter against which the older radars could not discriminate. This type of signal processing is very computational intensive and is performed in dedicated computers housed within the radar system itself. These computers take the sampled signal as input and output the finished results such as target parameters. Due to the computer's hardware configuration for high speed processing there is no easy method to gain access to the actual intermediate steps that are taking place in the long chain of signal processing functions.

A solution to this problem of student access to the intermediate steps is to simulate the signal processing on a personal computer. This simulation can be designed so as to show any step in the process that is desired and to allow the student to vary the inputs at will to compare different

algorithms. The simulation also calculates the performance characteristics particular to the process under consideration to allow comparisons between different algorithms. It is often easier for a student to comprehend visual outputs rather than analyze a table of performance characteristics, so the simulation will display the RSP results graphically.

B. FUNDAMENTALS OF RADAR SIGNAL PROCESSING

Older non coherent radars could detect only targets whose input signal to the radar was larger than the surrounding background interference. Smaller target signals hidden in the ground return, rain clutter and other interference were lost to the system since it was not possible to isolate the target signal from these interferences. This concept is shown in Figure 1 where only Target 1 could be detected by a non coherent radar. However, modern radars utilize the doppler effect to isolate the weak signal of a moving target from much larger background interference. This allows the detection of signals in interference that the older radars could not accomplish. Thus Targets 2 and 3 from Figure 1 could be detected by a modern radar. Implementation of this concept is shown in Figure 2 and its sub components are as follows

- The programmable signal processor (PSP) is the first computer used in the signal processing chain. It is designed as a high speed, computationally intensive tool

to process the signal from the receiver input (Rx).

- The A/D converter samples the signal at a rate on the order of 1 MHz.
- The adaptive nulling is used to cancel any interference coming in through the sidelobes of the radar's antenna.
- Pulse compression is used to compress the long transmitted pulses in order to have finer range resolution. This stage develops a processing gain equal to the pulse compression ratio (PCR) and thus builds up the target signal against background interference
- The clutter canceler significantly reduces ground clutter and other stationary forms of clutter using the moving target indicator (MTI) concept
- The doppler processor takes the spectrum of the incoming signal by performing fast Fourier transforms (FFT) on the signal samples. This process isolates the small target signal, which has undergone a doppler shift due to target velocity, from any stationary interference.
- The constant false alarm rate (CFAR) detector examines increases in signal amplitude above the average background interference in order to detect targets.
- The signal processing up to this point has been very computationally intensive and must be done in a high speed, optimized processor known as a programmable signal processor (PSP) in order to detect targets in real time.
- Data rates after the CFAR detector are much reduced and the processing is then done in a slower computer known as radar data processor (RDP). This processing is performed across several coherent pulse intervals (CPI's), and the end result is the target report which normally consists of the target's range, azimuth and velocity.

These signal processing steps are "invisible" to the student as there is no practical way to access these stages in either of the computers. This is the area which most needs simulation in order for students to become familiar with it.

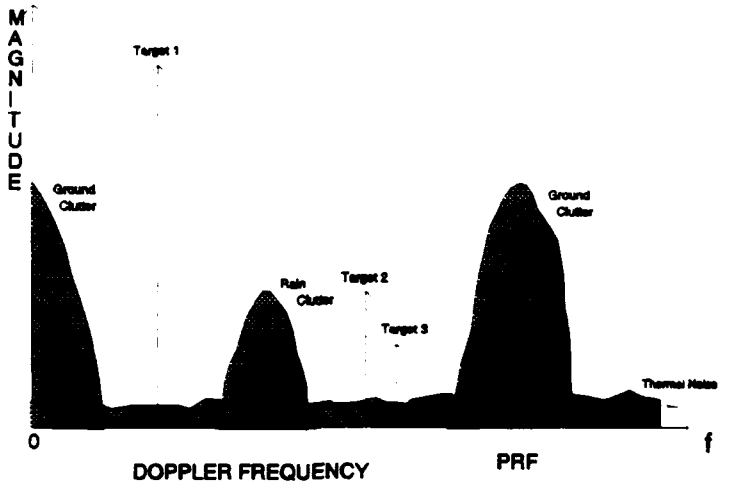


Figure 1 : Targets in clutter for a non MTI analog radar

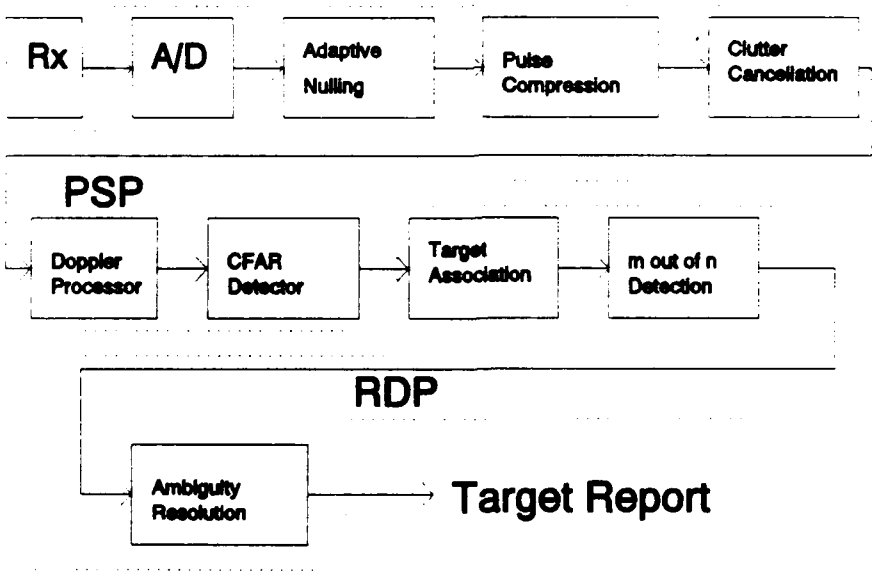


Figure 2 : Modern radar block diagram

C. TRADITIONAL METHODS AND SIMULATION

The overall subject of RSP consists of the underlying theory and implementation of digital signal processing algorithms. The particular problems inherent in the teaching of RSP in the classroom that can be overcome by the use of simulation are explained in this section.

1. Textbooks

The textbooks available in this area do a good job in explaining the basic principles of the subject. However, by necessity textbooks are limited to discussing only a few cases on the topic at hand due to limited text space. Even these few cases will require extensive computations which can not be shown in their entirety.

Simulation relieves the student of this computational burden. This leaves the student free to explore any aspects of the process which interests him. The student can experiment by changing the inputs or parameters of the problem and see their effects visually. Practical demonstrations on the application of the theory aids and motivates the student in comprehending the underlying principles of the process.

For example, in the section on pulse compression the student can easily find that not all signals will compress in a useable fashion, especially under doppler conditions. All he has to do is choose a signal and the doppler shift it

is to undergo, and the simulation will give the output for that particular case. The result is shown in graphical form and this brings the point home immediately.

2. Laboratory Hardware

In actual radar systems, as well as those set up in laboratories, the signal processing functions are performed by dedicated computers such as the PSP and the RDP. These computers pass the data to each other and sometimes send signals to other components of the radar. However, there is no easy way to access the outputs of each separate signal processing function. The intermediate results are almost completely inaccessible, even in a laboratory environment. Thus the most important parts of the RSP processes are hidden from the student.

This problem is also solved by using simulation programs to simulate each RSP process separately. This allows the student to separately examine each process, vary its inputs and examine the changes which occur within that process. He can then combine them for end to end simulation.

3. Numerical Computations

It is a fact of life that normal teaching methods lack the ability to study realistic radar problems. The following illustration on the detection of moving targets at a single azimuth angle explains why. A typical computation

would consist of perhaps 5 coherent processing intervals (CPI) for a MTI radar. Each CPI would typically consist of 1000 range bins for each of the 16 pulses constituting a CPI. Thus data for one azimuth angle consists of 5 x 1000 x 16 = 80,000 complex numbers, and each component of a complex number is typically represented by 12 bits. The doppler processing alone would require the computing of 5,000 sixteen point FFTs resulting in 80,000 range doppler cells. This doppler processing would be followed by a chain of algorithms and thus the processing for just one azimuth angle is well beyond a student's capability.

However, this signal processing can be readily simulated on a personal computer since simulation does not need to be done in real time as in an actual radar. It takes a few more seconds but produces results of the same accuracy as the real world system.

4. Simulation of RSP

There is one key point about the simulation of RSP algorithms which must be noted. The simulation will be just as accurate as the real system would be given the same inputs. The only difference between the real system and this simulation is that the generation of the input signals is done by the computer. All other algorithms and processes are performed in the same manner as they would be done in the PSP or RSP, albeit that the personal computer used for

these simulation programs is many orders of magnitude slower than the PSP.

D. REVIEW OF LITERATURE

Radar signal processing is a specialized application within the signal processing area. While RSP is an integral and critical part of a radar system, there are very few texts or works devoted solely to RSP due in part to the difficulty of discussing it using the traditional teaching methods. In general the best sources are either texts on general radar applications or those on signal processing.

In terms of the topics discussed in this thesis, the best source works were found to be the following:

- DFT as a filter bank [Ref. 1: Part V], [Ref. 3: chap 14], [Ref. 4: chap 7], [Ref. 6:p 119 to 125], [Ref. 7: chap 7], [Ref. 8:p 133 to 138].
- Weighting schemes for a DFT filter bank [Ref. 3: chap 10.8], [Ref. 6:p 426 to 427], [Ref. 7: chap 7 and 9], [Ref. 9: Appendix D].
- Pulse Compression [Ref. 1:Part IV], [Ref. 2:chap 2], [Ref. 3:chap 4 and 10], [Ref. 4:chap 8], [Ref. 5:chap7].

The above list is by no means complete, and better sources for individual topics may exist. The thesis in itself should provide a good introduction to these topics for a student in RSP. In particular the simulation programs will allow him to manipulate the RSP algorithms to gain a fuller understanding of the processes involved.

II. THE DISCRETE FOURIER TRANSFORM AS A FILTER BANK

Modern coherent radars discriminate between targets and clutter on the basis of target motion. Moving targets receive a doppler frequency shift that causes the frequency of the received target return to be different from the stationary clutter return. Signal processing methods that take advantage of the frequency shift for moving target indicator (MTI) radars are the delay line canceler and the discrete Fourier transform (DFT) filter bank.

The delay line canceler for MTI radar is reviewed and the advantages of the DFT filter over it are highlighted. The development of the DFT as a filter bank for radar applications is addressed more thoroughly.

A. DELAY LINE CANCELER VERSUS DFT FILTER BANK

The delay line canceler used in MTI radars is a time domain filter. It rejects the clutter at low frequencies and passes the target signal at higher frequencies. It is essentially a band reject filter with the notch placed at dominant clutter frequencies. [Ref. 6:p107] One simple way of doing this is by storing the input value for a period of time equal to the PRI and then subtracting the stored value from the new value. This removes the stationary clutter at low doppler frequencies. However, it also cancels any

signals at frequencies that are close to multiples of the PRF. Figure 3 shows the spectrum of a radar return with ground clutter, rain clutter and targets at one range interval. The radar can not separate out the return by way of the doppler frequencies and thus processes all of the frequencies present in this interval from zero to the PRF. The ground and rain clutter are present at all times for this range and thus the constant false alarm rate (CFAR) mechanism raises the detection threshold to a level that requires a greater target magnitude response than the average clutter level. The detection threshold for this situation is shown in Figure 3. Only Target 1 could be detected as it is the only target that exceeds the background clutter level.

Figure 4 shows the spectrum at the output of a delay line canceler where the ground clutter has been attenuated. The radar still processes all of the frequencies present in this interval from zero to the PRF, but the magnitude of the stationary clutter has been attenuated so the CFAR detection level is much lower than in Figure 3. However, the delay line canceler has no effect on the rain clutter, and Target 2 is still masked by the rain clutter since the CFAR detection level is now dictated by the rain clutter magnitude. However, if a DFT filter bank were used, then the rain clutter can be isolated from target 2. This is shown in the spectrum of the return for an eight filter DFT

bank at Figure 5. Now the rain clutter is in a separate filter from the return of Target 2. Thus Target 2 could now

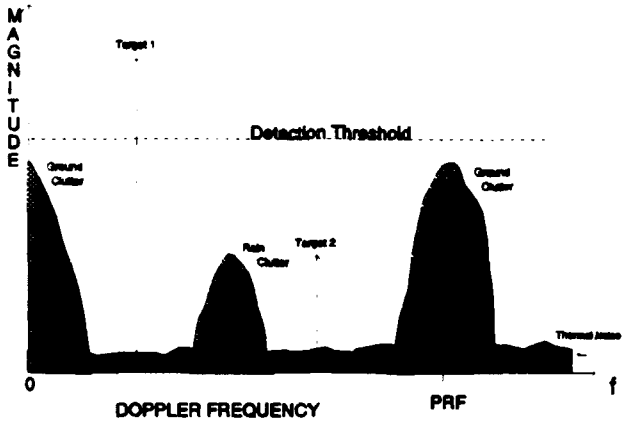


Figure 3 : True spectrum of a radar return showing the detection threshold set by CFAR

be detected as the CFAR is set individually for each filter and Target 2 is significantly greater than the thermal noise in its filter. The main advantages of a DFT filter bank over a delay line canceler are: [Ref. 6:p107 to 117], [Ref.5:p7 to 33] and [Ref.12]

- The spectrum of the return is divided into several different filters to better separate clutter from targets.
- Also, it is possible to change the filter width by varying the PRF or N so that any targets that appear near clutter frequencies for one PRF can be moved to a filter that has very little clutter in it for a second PRF. This will be demonstrated more clearly in the simulation program in Chapter III.

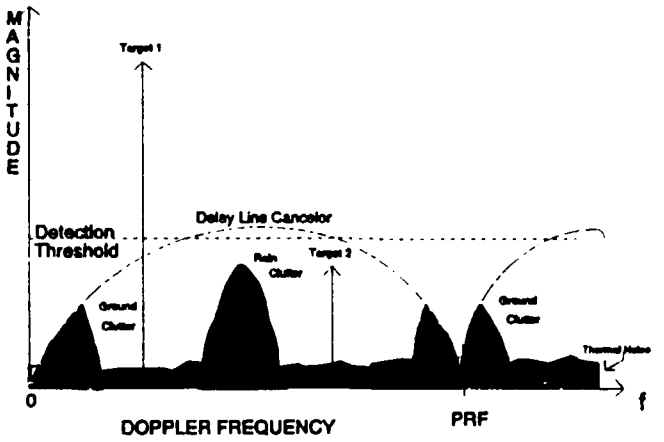


Figure 4: Spectrum of a radar return at the output of a delay line canceler. The ground return clutter and any returns at frequencies that are near multiples of the PRF have been canceled.

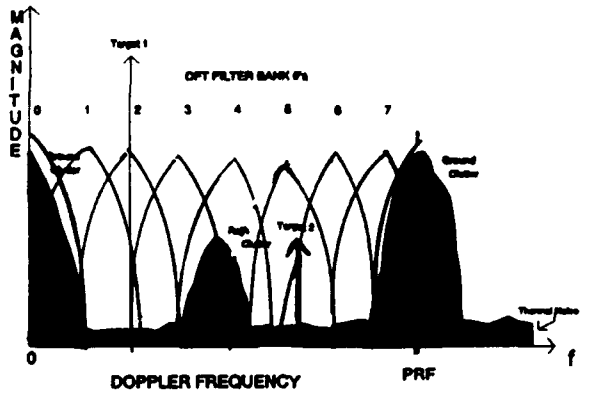


Figure 5 : Spectrum of radar return after a DFT filter bank. The ground clutter is in filters 0 and 1, while the rain clutter is in filters 3 and 4. Target 1 is in filter 2 and Target 2 is in filter 5.

- Receiver noise is limited to one filter width and thus the signal to noise (SNR) level is raised for a target return.
- The possibility exists to measure the target's doppler frequency and thus compute its radial velocity.

B. DFT FILTER BANK FOR COMPLEX SINUSOIDAL INPUT

The development of the DFT as a filter bank in radar applications is presented for a complex sinusoidal input and for a real sinusoidal input. A complex sinusoidal input would result from the use of quadrature sampling of an input signal whereas a real input would be from a single channel sampling process. Figures 6 and 7 show how these inputs are formed. The detector in these figures consists of a mixer followed by a low pass filter. The main advantages of quadrature sampling over single channel sampling are:

- Positive and negative doppler frequencies can be distinguished.
- The blind phase problem can be avoided in quadrature sampling. Blind phases may occur in single channel sampling when the sampling interval is such that it coincides with the zero crossing of the incoming signal. For example, when $\phi \approx 90°$, $\cos(\phi)$ is a very small number for single channel sampling. In quadrature sampling these blind phases are eliminated by the presence of $\sin(\phi)$, which will be high when $\cos(\phi)$ is low. [Ref. 3:p15.7], [Ref. 5:p7 to 40].

The development of the DFT for a quadrature sampled signal follows.

1. The Input Signal

If the input signal is a complex sinusoid it can be represented by

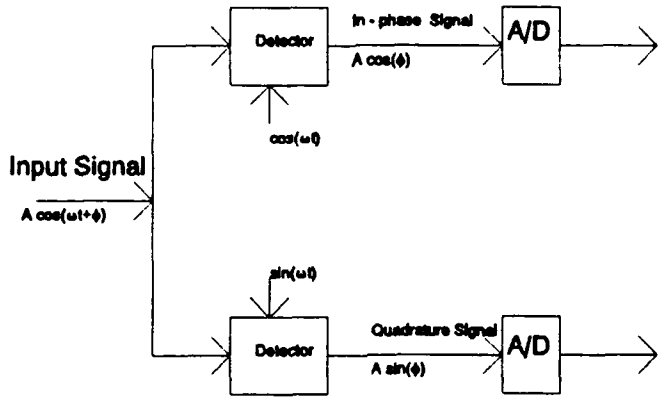


Figure 6 : Quadrature Sampling which results in a complex input signal

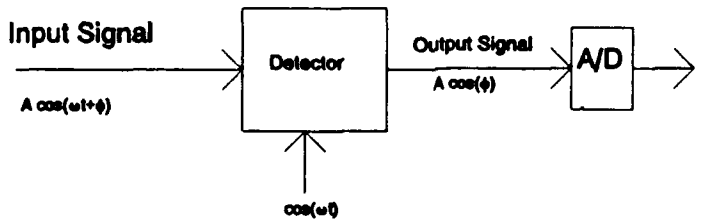


Figure 7 : Single Channel Sampling which results in a real input sinusoid

$$s(t) = \cos(2\pi ft) + j\sin(2\pi ft) = e^{j2\pi ft} \quad (1)$$

Once it has undergone quadrature sampling it can be represented as a discrete signal by noting that x = fNT and t = nT so that

$$s(nT) = e^{j2\pi \frac{x}{NT} nT} = e^{j\frac{2\pi}{N} xn}. \quad (2)$$

where

- x is the discrete frequency.
- N is the total number of samples.
- n is the sub sample number, n = 0:N-1, $n \epsilon I$.
- PRF is the pulse repetition frequency. This is the sampling frequency for radar applications.
- nT is the discrete sample time.

2. Discrete Fourier Transform Development

The DFT for a finite length sequence s(n) is as follows

$$F(k) = \sum_{n=0}^{N-1} s(n)\, e^{-j\frac{2\pi nk}{N}}. \quad (3)$$

Substituting for s(n) from Equation 2 leads to the DFT response for a complex sinusoidal input

$$F(k) = \sum_{n=0}^{N-1} e^{j\frac{2\pi n(x-k)}{N}} = \sum_{n=0}^{N-1} e^{jn\theta}, \quad (4)$$

$$where\ \theta = \frac{2\pi(x-k)}{N}. \qquad (5)$$

Then using the mathematical relation for the summation of a finite geometric series, Equation 5 can be rewritten as

$$\begin{aligned} F(k) &= \frac{1-e^{jN\theta}}{1-e^{j\theta}} \\ &= \frac{e^{-j\frac{N\theta}{2}}}{e^{-j\frac{\theta}{2}}} * \frac{e^{j\frac{N\theta}{2}}-e^{-j\frac{N\theta}{2}}}{e^{j\frac{\theta}{2}}-e^{-j\frac{\theta}{2}}} \\ &= e^{-j(N-1)\frac{\theta}{2}} * \frac{\sin(\frac{N\theta}{2})}{\sin(\frac{\theta}{2})} \end{aligned} \qquad (6)$$

3. Magnitude Response of the DFT to a Complex Input

In the detection of targets only the magnitude at the filter output is considered, and thus phase can be ignored. The magnitude of the output of the DFT will be

$$|F(k)| = \left|\frac{\sin(\frac{N\theta}{2})}{\sin(\frac{\theta}{2})}\right|, \qquad (7)$$

and substituting for θ from Equation 5 leads to,

$$|F(k)| = \left|\frac{\sin(\pi(x-k))}{\sin(\frac{\pi(x-k)}{N})}\right|. \qquad (8)$$

The frequency response of the DFT to a quadrature sampled input sinusoid is shown in Figure 8. This plot is of the magnitude response of Equation 8 for k = 0 over a frequency range of several PRFs. The magnitude response pattern of F(k) is repetitive in nature and repeats itself every PRF because the PRF is the sampling frequency. This then leads to the conclusion that F(k)=F(k±N). This is proven as follows:

$$|F(k)| = |F(k\pm N)| \tag{9}$$

$$= \left|\frac{\sin(\pi(x-(k\pm N))}{\sin(\frac{\pi(x-(k\pm N)}{N})}\right| \tag{10}$$

$$= \left|\frac{\sin(\pi(x-k)\pm\pi N)}{\sin(\frac{\pi(x-k)}{N}\pm\pi)}\right| \tag{11}$$

$$= \left|\frac{\sin(\pi(x-k))}{\sin(\frac{\pi(x-k)}{N})}\right| \tag{12}$$

$$= |F(k)| \qquad QED \tag{13}$$

The same response for k = 0 is shown in Figure 9 over one PRF from the SINGPLOT option of the DFTBANK simulation program found in Chapter III. This pattern is considered to be the 0^{th} filter (ie. filter # k=0) of a DFT filter bank. If the magnitude response of this filter crosses the detection threshold, then a target is reported as being within the frequency range of the mainlobe of the

magnitude response for that filter. Should a target in the sidelobes of the filter exceed the threshold value, then the radar will assume it is coming from within the mainlobe and thus a false target frequency will be declared.

The null to null frequency width of the filter is noted on Figure 9, as is the 3 dB frequency width. The equations to determine these widths are

$$\Delta f_{null} = Null\ to\ Null\ MAINLOBE\ WIDTH = \frac{2PRF}{N} \qquad (14)$$

and

$$\Delta f_{3dB} = 3\ dB\ MAIN\ LOBE\ WIDTH \approx 0.89\frac{PRF}{N}. \qquad (15)$$

By varying k from 0 to N-1, kϵI the DFT filter bank is formed from the mainlobes of the N filters. This is shown in Figure 10 from the MULTMAIN option of the DFTBANK simulation program. The straddle loss is defined as the loss in magnitude response between the peak response of the filter and the response of the mainlobe when it crosses the mainlobe of the adjacent mainlobe. The frequency range for mainlobe detection of a filter is defined as being from one adjacent crossover point to another. This is shown in

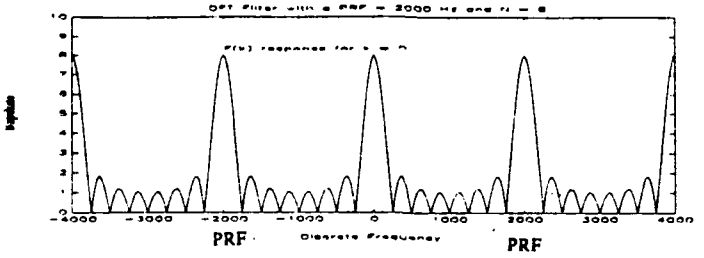


Figure 8 : F(k) Magnitude Response for k = 0 over several PRFs. The pattern repeats itself every PRF.

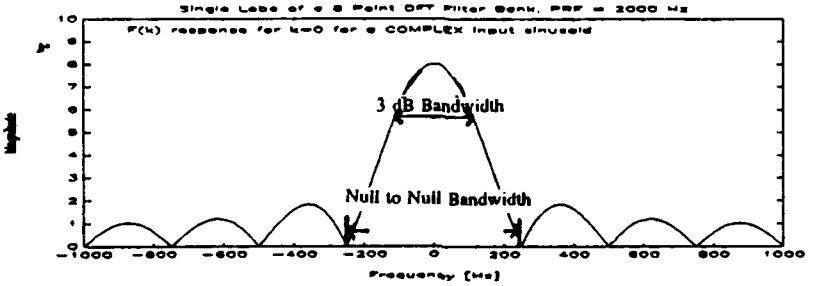


Figure 10 : F(k) Magnitude Response for k = 0 using the SINGPLOT option, PRF = 2000 Hz and N =8.

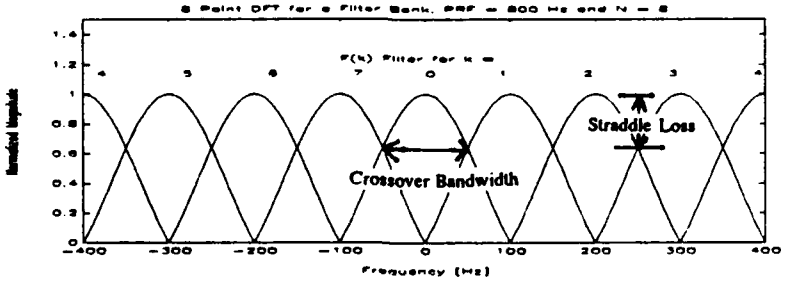


Figure 9 : DFT filter bank showing filter mainlobe crossovers for PRF = 2000 Hz and N=8 from MULTMAIN option of DFTBANK simulation.

Figure 10. Thus the crossover point defines the worst response a target would have while within a filter bank.

$$\Delta f = CROSSOVER\ MAINLOBE\ WIDTH \approx \frac{PRF}{N} \tag{16}$$

C. DFT FILTER BANKS FOR REAL SINUSOIDAL INPUT

1. The Input Signal

If the input signal is a real sinusoid then it can be represented by

$$s(t) = \cos(2\pi ft) = \frac{1}{2}[e^{j2\pi ft} + e^{-j2\pi ft}]. \tag{17}$$

The sampled signal can be represented in discrete form by

$$s(n) = \frac{1}{2}[e^{j\frac{2\pi}{N}xn} + e^{-j\frac{2\pi}{N}xn}]. \tag{18}$$

2. Discrete Fourier Transform Development

The form of the DFT for the real finite length sequence s(n) is the same as in the complex case. Thus substituting Equation 18 into Equation 3 leads to

$$F(k) = \sum_{n=0}^{N-1} \frac{1}{2}[e^{j\frac{2\pi n(x-k)}{N}} + e^{-j\frac{2\pi n(x+k)}{N}}]. \tag{19}$$

Given the duality inherent in Equation 19 it is evident that F(k)=F(-k). Using the mathematical relation for the convergence of a finite geometric series in the same fashion as for the complex sinusoid results in

$$F(k) = \frac{1}{2}\left[e^{-j(N-1)\frac{\theta}{2}} * \frac{\sin(\frac{N\theta}{2})}{\sin(\frac{\theta}{2})} + e^{-j(N-1)\frac{\phi}{2}} * \frac{\sin(\frac{N\phi}{2})}{\sin(\frac{\phi}{2})}\right], \quad (20)$$

$$where\ \theta = \frac{2\pi(x-k)}{N}, \quad (21)$$

$$and\ \ \phi = \frac{-2\pi(x+k)}{N}. \quad (22)$$

3. Magnitude of the DFT Response to a Real Input

Rewriting Equation 20 leads to:

$$F(k) = \frac{1}{2} * e^{-j(N-1)\frac{\theta}{2}}\left[\frac{\sin(\frac{N\theta}{2})}{\sin(\frac{\theta}{2})} + e^{-j(N-1)\frac{(\phi-\theta)}{2}} \frac{\sin(\frac{N\phi}{2})}{\sin(\frac{\phi}{2})}\right] \quad (23)$$

Therefore the magnitude response is

$$\therefore \quad |F(k)| = \frac{1}{2} * \left| \frac{\sin(\frac{N\theta}{2})}{\sin(\frac{\theta}{2})} + e^{-j(N-1)\frac{(\phi-\theta)}{2}} \frac{\sin(\frac{N\phi}{2})}{\sin(\frac{\phi}{2})} \right|. \tag{24}$$

The remaining exponential term can be reduced by substituting in for θ and ϕ from Equations 21 and 22 so

$$\begin{aligned} e^{\frac{-j(N-1)}{2}(\phi-\theta)} &= e^{\frac{-j(N-1)}{2}\frac{2\pi}{N}((x+k)-(x-k))} \\ &= e^{\frac{-j(N-1)}{N}2\pi k} \\ &= e^{-j2\pi k} * e^{j\frac{2\pi k}{N}} \\ &= e^{j\frac{2\pi k}{N}}. \end{aligned} \tag{25}$$

Substituting Equations 25 into Equation 24 thus gives a magnitude response of

$$|F(k)| = \frac{1}{2} * \left| \frac{\sin(\pi(x-k))}{\sin(\frac{\pi(x-k)}{N})} + e^{j\frac{2\pi k}{N}} * \frac{\sin(\pi(x+k))}{\sin(\frac{\pi(x+k)}{N})} \right|. \tag{26}$$

The frequency response of the DFT for a single channel sampled input sinusoid is shown in Figure 11. This plot is of the magnitude response of Equation 26 for k = 1 using the SINGPLOT option from the DFTBANK simulation program at Chapter III. For k=1 there are two equal magnitude mainlobes spaced evenly about a centre frequency that is a multiple of the PRF. The obvious difference between the complex and real input data cases is that the former has a single main lobe and the real sinusoid has two main lobes that are equally offset from the PRF. In Figure 12 for k=-1 the magnitude response of the DFT is identical to that of Figure 11. This is as expected given the duality of Equation 19 such that F(k)=F(-k). Therefore it is not possible to distinguish positive and negative frequencies of the same magnitude since they will have an identical magnitude response from the DFT filter bank.

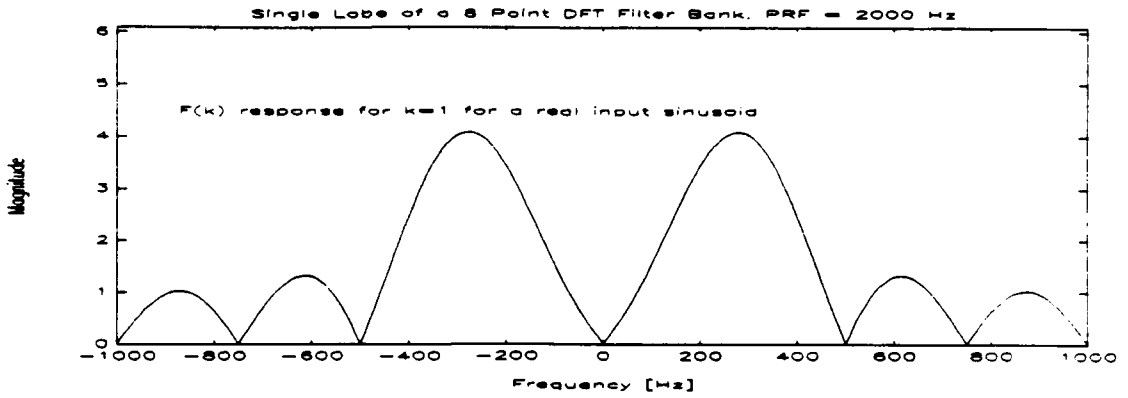


Figure 11 : F(k) magnitude response of a real input sinusoid for k=1 using the SINGPLOT option, PRF = 2000 Hz and N=8.

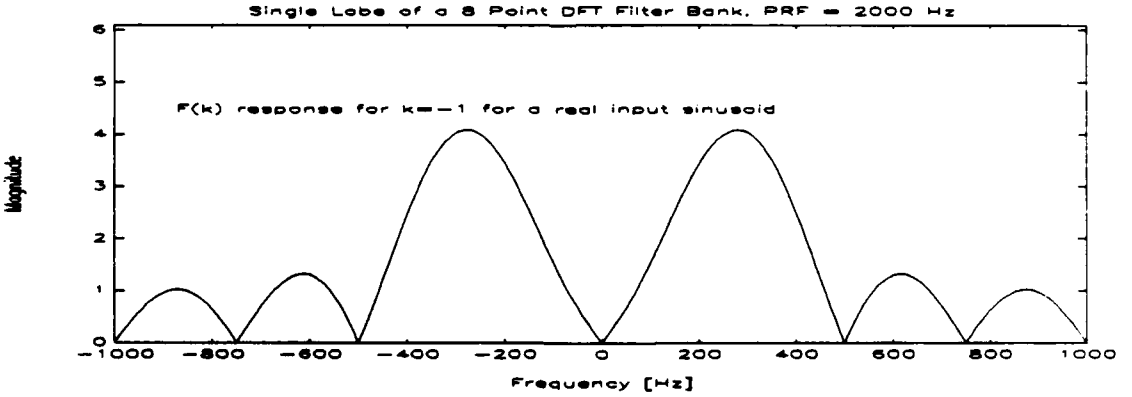


Figure 12 : F(k) magnitude response of a real input sinusoid for k=-1 using the SINGPLOT option, PRF = 2000 Hz and N=8.

III. SIMULATION OF A DFT FILTER BANK

This chapter describes the methodology of the simulation of a DFT filter bank response to a real or complex sinusoid, presents an overview of the program structure, describes how to use the program and shows how to interpret the results.

The DFT filter bank simulation program implements a filter bank showing the magnitude response of a real or complex sinusoidal input for all unambiguous frequencies. The following parameters can be varied in the simulation;

- N = the number of points for which the DFT is to be calculated,
- PRF = the pulse repetition frequency of the radar,
- k = the filter designator (if required), and
- f_{tgt} = the doppler frequency of the target (if required).

By controlling these parameters the student can construct DFT filter banks for inputs that are real or complex sinusoids. Some of the situations that can be addressed are:

- The response of a filter bank to a target appearing at a given doppler frequency (f_{tgt}) can be assessed. If the target's doppler frequency is an ambiguous frequency of the filter bank, then the alias target frequency (f_a) report is given. The alias target frequency is developed from

$$f_a = f_{tgt}\ modulo(PRF) \tag{27}$$

- If the target and clutter are in the same filter at a given PRF and N, then the filter bank can be changed by varying either N or the PRF to move the target and clutter into separate filters.
- The response of a DFT filter bank using single channel or quadrature sampling to positive or negative doppler targets can be observed.
- There are numerous other possibilities left to the user's initiative.

A. PURPOSE OF THE SIMULATION PROGRAM

The equations implemented by the subprograms are for a real input sinusoid from single channel sampling and a complex sinusoid from quadrature sampling as seen in Figures 6 and 7. The frequency range of the filter bank is for

$$|f| \leq \frac{PRF}{2} \tag{28}$$

which covers all of the unambiguous frequencies for that filter bank. The frequency is stepped in increments of 0.005*PRF which gives a smooth plot while maintaining a reasonable processing speed on a DOS 386 machine.

B. PROGRAM STRUCTURE

The overall program structure can be seen in Figure 13. The structure is that of a main program (DFTBANK) which calls various subprograms (SINGLOBE, MULTMAIN, ... etc) as required. There is an error message (ERRORMSG) which

detects an improper data entry, notifies the operator and restarts the process. It is possible to exit the simulation from either the main or subprograms. This structure was used since it allows maximum operator control and input. It is a robust format that is data entry error tolerant. The

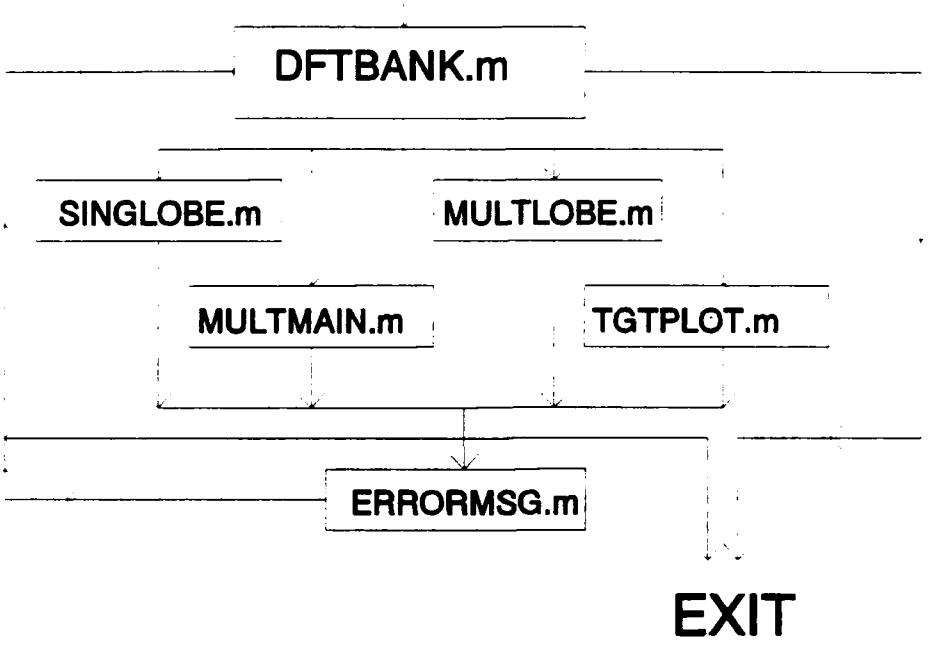


Figure 13 : Overall program structure for DFT bank simulation

other advantage of this format is that if an operator desires to change one of the subprograms or add another subprogram the system can accommodate this. Any errors made in this process will affect only the subprogram in question. The rest of the simulation will remain operational while the new addition is being debugged.

C. OPERATING INSTRUCTIONS AND GUIDE

1. Initial Setup

To use this simulation the following MATLAB.m files must be copied into the user sub directory. Copies of these programs are available from Dr GS Gill at USNPGS.

- DFTBANK.m is the simulation control program.
- SINGLOBE.m is the simulation subprogram responsible for displaying the magnitude response of a single filter across all unambiguous frequencies in the filter bank.
- MULTMAIN.m is the simulation subprogram responsible for displaying the mainlobe magnitude response of all filters in the filter bank for all unambiguous frequencies.
- MULTLOBE.m is the simulation subprogram responsible for displaying the full magnitude response for all filters in the filter bank for all unambiguous frequencies.
- TGTPLOT.m is the simulation subprogram responsible for calculating and displaying the filter response for a target at a given doppler frequency. It can accept ambiguous target frequencies and the filter response will be for the alias frequency. The filter that this alias frequency lies in can result in a positive doppler being identified as negative or vice versa.
- ERRORMSG.m informs the operator when an entry error has been made and restarts DFTBANK after clearing the system of all variables.

This simulation works on a DOS 286 machine, but a 386 or higher is preferable for increased speed. To start the simulation type DFTBANK [ENTER] at the MATLAB prompt.

2. Printing Graphical Outputs

The assumption has been made that the PRINT SCREEN key will be used to print out plots. To use this key the

DOS graphics command must have been issued at the DOS prompt before any printing is attempted; C:\>> graphics [ENTER] is the necessary command.

Should it be preferable to create meta files for each plot, the operator has two options:

- Modify all plot options in the subprograms by adding a meta command and file name after the plot command, or
- Press the Ctrl-C keys to exit the simulation once the plot is on the screen. This will return him to the MATLAB prompt and he can issue the meta command from there. To restart the simulation type DFTBANK [ENTER].

3. Program Options

After the DFTBANK command has been issued to start the simulation the first screen seen by the operator is at Figure 14. The second screen the operator will see is at Figure 15. At this point the operator must choose whether he wishes to process a real or complex sinusoid for the simulation. Then he has to choose the type of system response that he wants to manipulate. Once this has been done the main program transfers control to the subprogram selected in Figure 16. All of the screens for the subprograms are very similar, the only difference being whether "k" or "f_{tgt}" are required inputs.

The choice for a dB or ratio plot determines whether the magnitude scale on the y-axis will be plotted for ratio values or in a decibel format. The dB format is preferable when determining the sidelobe level response. For a more

```
   This program will allow you to create Doppler Filter
Banks for either real or complex input sinusoids.  Real
sinusoids are the result of single channel sampling while
complex sinusoids are from quadrature sampling.  Single
channel sampling means that it will be impossible to
differentiate between positive or negative doppler.
Quadrature sampling allows positive or negative doppler
to be distinguished.

   The options are to see the full response of a single
filter over the full spectrum of unambiguous frequencies,
to see the mainlobe response of a DFT bank of filters, to
see the full response of a DFT bank of filters (including
sidelobes) or to see the magnitude response of each
filter in a DFT bank to a target occurring at a
particular frequency.

          TOUCH ANY KEY TO CONTINUE
```

Figure 14: First Screen of DFTBANK main control program

```
   You must choose whether to process a real sinusoid
from single channel sampling or a complex sinusoid from
quadrature sampling.

   FOR A REAL SINUSOID, choose 1

   FOR A COMPLEX SINUSOID, choose 2

   ENTER YOUR CHOICE NOW,  SINUSOID =

  You must choose which type of response you wish to see
  To see a single filter response, ENTER 1
  To see a filter bank with main lobe response,ENTER 2
  To see a filter bank with side lobe response,ENTER 3
  To see target values calculated for each filter,ENTER 4
  TO EXIT THIS PROGRAM,ENTER 0
 ENTER YOUR CHOICE NOW
 CHOICE =
```

Figure 15: Second Screen of DFTBANK main control program

general overview, the ratio format is better. The actual inputs for the variables described in Figure 16 are given on the next screen at Figure 17.

```
  YOU MUST NOW ENTER SOME OF THE NECESSARY PARAMETERS.
  These parameters are the number 'N' of DFT points the
filter is to calculate.  This is also the number of
filters there will be in the filter bank.
   Next you must choose the PRF you wish to consider.
   Then you must choose the 'k' offset for the DFT to
determine which filter in the DFT filter bank you will
select.  The choice of 'k' is an integer value usually
from 0 to N-1, but you are not limited to this selection
if you desire to see the periodic nature of the DFT.
   Finally you must choose whether you wish the display
of the magnitude response to the DFT to be in normalized
dB or ratio form.

%  TOUCH ANY KEY TO CONTINUE
```

Figure 16: First Screen of a subprogram

```
  Number of DFT points you wish is,  N =
  The PRF in Hz you wish is,  PRF =
  The Filter K # you wish to use is,  k =
  For a dB plot choose 1, for a ratio plot choose 2,  ? =
```

Figure 17: Second Screen of a subprogram input request

Once all of the graphical outputs and results have been shown, each subprogram will terminate with the screen at Figure 18 where the operator can choose to do the same subprogram for the same type of input to the DFT (the SINUSOID variable remains constant), return to the main program (DFTBANK) to try a different simulation or exit the simulation.

Any data entry error that the inherent MATLAB error routines do not recognize will be detected by the simulation and the error message at Figure 19 will be displayed. All variables will be cleared and the main program DFTBANK will be restarted.

```
  YOU MUST NOW CHOOSE TO EITHER DO THIS CHOICE AGAIN,
RETURN TO THE MAIN MENU OR EXIT THE SIMULATION
   DO THE SINGLE LOBE AGAIN,          CHOOSE 1
   TO RETURN TO THE MAIN MENU,        CHOOSE 2
   TO EXIT THE SIMULATION,            CHOOSE 0
```

Figure 18 : Final Screen of a subprogram

```
   YOUR CHOICE DOES NOT CORRESPOND TO THOSE AVAILABLE.
   THE PROGRAM WILL BE RESTARTED FOR YOU.
   REMEMBER YOU MUST CHOOSE 0 TO EXIT.

   PRESS ENTER TO CONTINUE...
```

Figure 19 : Error message subprogram screen

D. INTERPRETATION OF RESULTS

The output of the simulation is mostly graphical in nature since graphical displays are readily assimilated and provide a more intuitive understanding of the process. Results from each of the subprograms are presented. These plots are analyzed to demonstrate some of the capabilities of the simulation program and to provide worked examples as a reference guide. In each case the data inputs used are provided and the selection process described.

1. Single Filter Response

The SINGLOBE subprogram displays the magnitude response of a single DFT filter across all unambiguous frequencies in the filter bank. These unambiguous frequencies are for $|f| \leq PRF/2$. While it is the mainlobe magnitude response of the filter that defines the frequency coverage of the filter, there will be some response to any frequency present in the filter bank. If the response for a frequency were from outside of the mainlobe magnitude response range and it exceeded the detection threshold set for that filter, an additional target report would be generated.

The operator can vary the PRF and N in the simulation to achieve a filter response with an acceptable filter width, frequency resolution and doppler frequency coverage. In the following example the inputs to the simulation were:

- SINUSOID = 1 for a real input sinusoid to be processed,
- CHOICE = 1 for a single filter response,
- N = 8,
- PRF = 2000 Hz,
- k = 1 for the F(1) magnitude response, and
- dB_ratio = 2 for a ratio plot.

All of these inputs will stay the same for all subsequent simulations in this section unless noted

otherwise. The plot resulting from these simulation choices is at Figure 20. The twin mainlobe response for a real input sinusoid is clearly evident. Had dB_ratio = 1 been

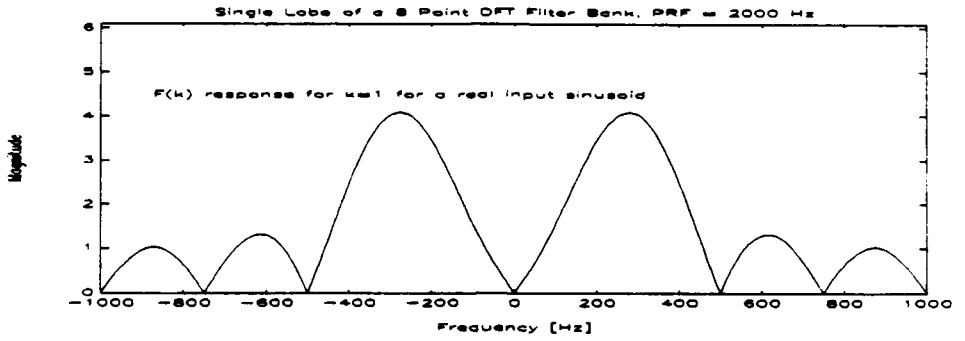


Figure 20 : Magnitude response of a real input sinusoid for F(k) in a DFT filter bank where N=8, PRF = 2 KHz and k=1

selected than the dB ratio magnitude response at Figure 21 would have been plotted. Selecting k=-1 would result in Figure 22. This shows why positive and negative doppler can not be distinguished by a real input DFT filter bank as F(k)=F(-k) so that targets at the same positive or negative doppler frequency will have the same response.

Selecting k=7 would result in Figure 23, which shows the recurring nature of the DFT as F(k)=F(k±N) with respect to Figure 22 where k=-1. Had SINUSOID=2 for a complex sinusoid from quadrature sampling been chosen then the graph at Figure 24 would have been plotted. The differences between real and complex input sinusoids are evident.

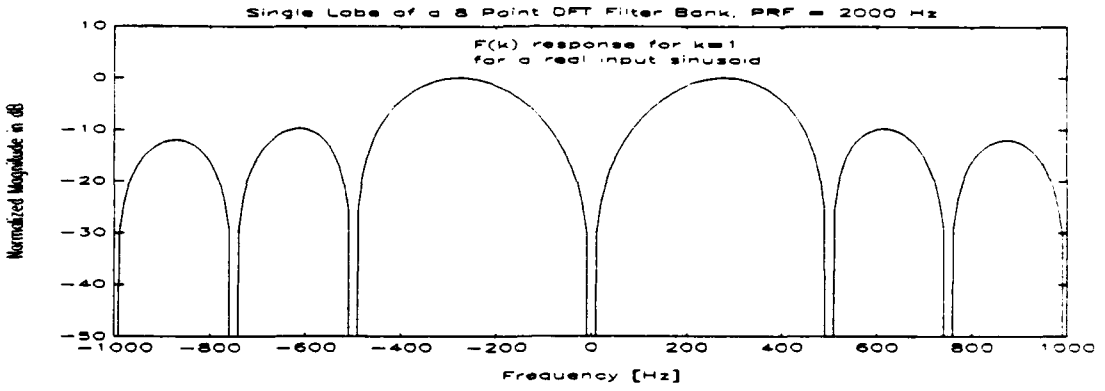


Figure 21 : Magnitude response in dB of a real input sinusoid for F(k) in a DFT filter bank where N=8, PRF = 2 KHz and k=1

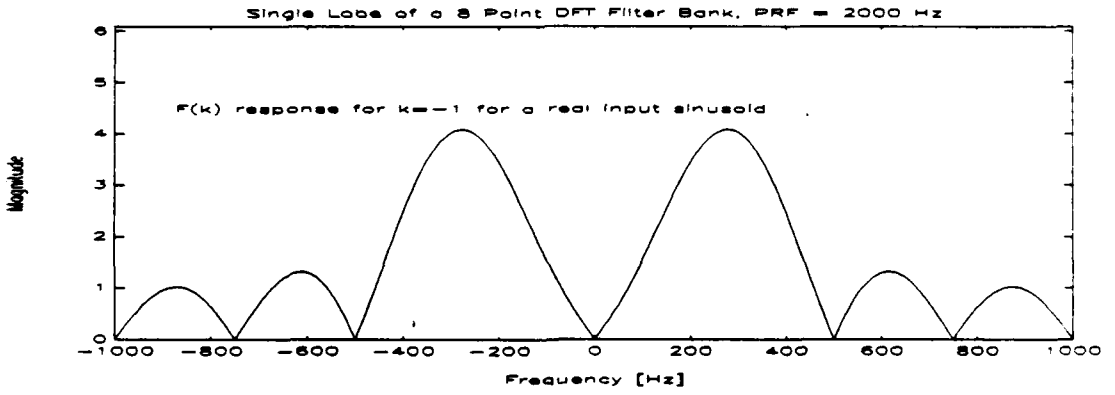


Figure 22 : Magnitude response of a real input sinusoid for F(k) in a DFT filter bank where N=8, PRF = 2 KHz and k=-1.

2. Mainlobe Filter Response

The MULTMAIN subprogram displays the mainlobe magnitude response of all filters in the filter bank for all unambiguous frequencies. The choices are the same as for SINGLOBE, with the exception of "k" since the mainlobe magnitude responses for k=0 to N-1, k ϵ I are all plotted automatically.

In the following example the inputs to the simulation were

- SINUSOID = 2 for a complex input sinusoid,
- CHOICE = 2 for a filter bank with mainlobe response,
- N = 8,
- PRF = 2000 Hz, and
- dB_ratio = 2 for a ratio plot.

All of these inputs will stay the same for all subsequent simulations unless noted otherwise. The plot resulting from these simulation choices is at Figure 25. The crossover filter width, straddle loss and other performance measurements can be noted from it. Should SINUSOID = 1 have been chosen then the plot at Figure 26 would have resulted. The different filter bank peak responses to a real input sinusoid for the various k inputs of Equation 34 can be seen. Therefore the detection thresholds for the different filters must vary to account for this effect.

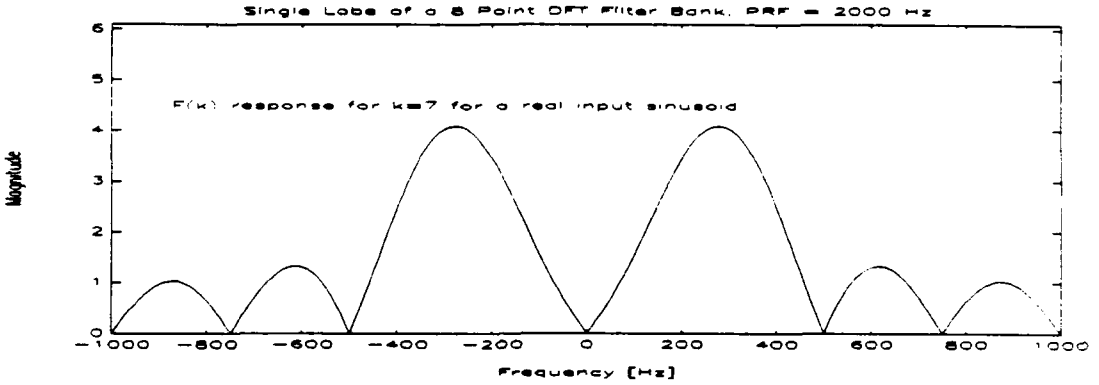


Figure 23 : Magnitude response of a real input sinusoid for F(k) in a DFT filter bank where N=8, PRF=2 KHz and k=7

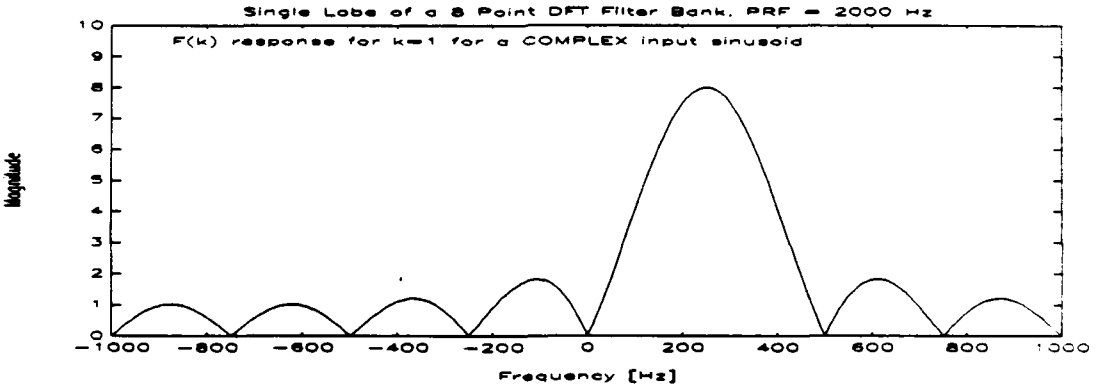


Figure 24 : Magnitude response of a complex input sinusoid for F(k) in a DFT filter bank where N=8, PRF=2 KHz and k=1

3. DFT Filter Response With Sidelobes

The MULTLOBE subprogram is the same as the MULTMAIN one, except that the full response for each filter is calculated and plotted, not just the mainlobes.

In the following example the inputs to the simulation were

- SINUSOID = 2 for a complex input sinusoid,
- CHOICE = 3 for a filter bank with side lobe response,
- N = 8,
- PRF = 2000 Hz, and
- dB_ratio = 2 for a ratio plot.

All of these inputs will stay the same for all subsequent simulations unless noted otherwise. The plot resulting from these simulation choices is at Figure 27. This demonstrates where the various filters have a mainlobe crossover and the peak sidelobe levels. From it the choice of where to set the detection threshold can be based on the following;

- to avoid declaring targets in more than one mainlobe of a filter, the threshold must be set higher than the mainlobe crossover point, or
- to avoid declaring targets from the sidelobe of a filter, the threshold must be set higher than the highest sidelobe.

This gives the operator a good feel for the dynamic range over which he can detect targets that vary in signal strength and not have excessive detection threshold

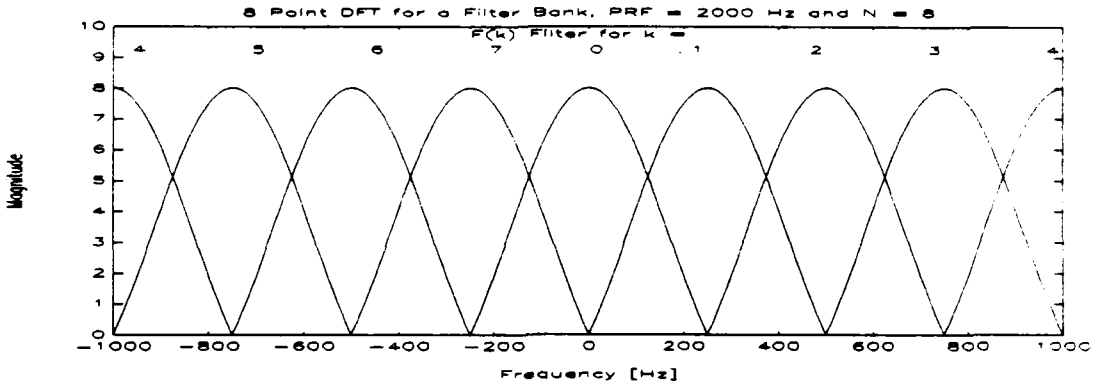


Figure 25 : Magnitude response of a complex input sinusoid for F(k) in a DFT filter bank where N=8, PRF=2 KHz and k=0 to N-1, $k \epsilon I$

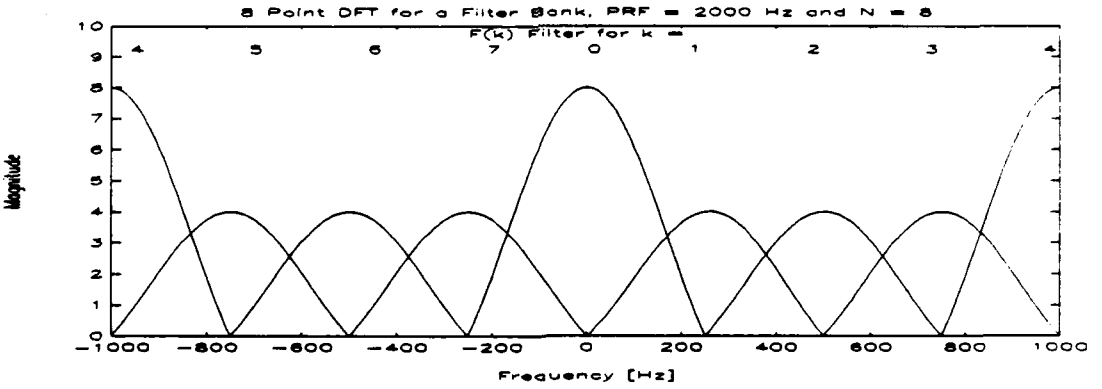


Figure 26 : Magnitude response of a real input sinusoid for F(k) in a DFT filter bank where N=8, PRF=2 KHz and k=0 to N-1, $k \epsilon I$

crossings and false doppler reports. A measurement of this nature is easier to do if the dB_ratio = 2 for a dB plot has been selected as in Figure 28.

4. Filter Bank Response to a Specific Target Frequency

The TGTPLOT subprogram is the same as the MULTLOBE subpogram except that it will require an additional input for the doppler frequency the target or clutter is to appear at and that all plots are automatically of a dB nature. If the doppler frequency given does not satisfy the Nyquist criteria then an alias frequency (f_a) will be calculated from

$$f_a = f_{tgt}\ modulo(PRF)\,. \qquad (29)$$

This alias frequency is then processed by the filter bank. The filter that the alias frequency lies in can result in a positive doppler being identified as negative or vice versa.

In the following example the inputs to the simulation were:

- SINUSOID = 2 for a complex input sinusoid,
- CHOICE = 4 for a target plot,
- N = 8
- PRF = 2000 Hz, and
- f_{tgt}= 425 Hz

The plot resulting from these simulation choices is at Figure 29. The plot of Figure 30 shows how a f_{tgt}= 5250 Hz

that does not satisfy the Nyquist criteria can result in a f_s that gives misleading results.

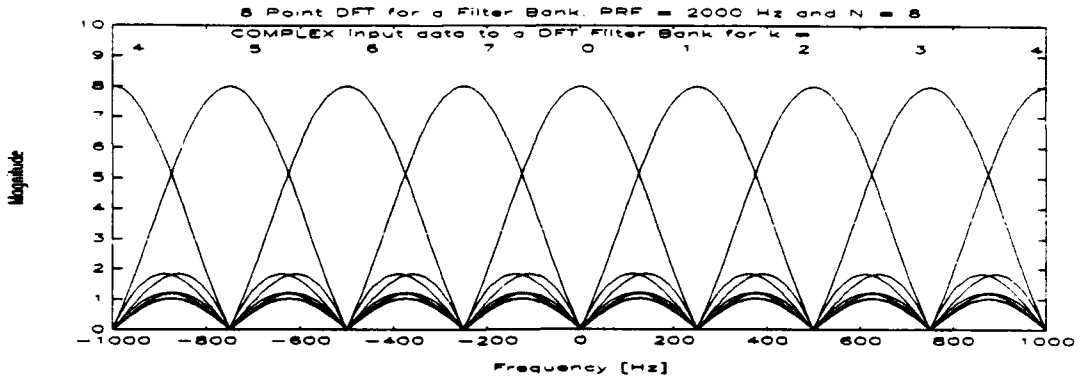


Figure 27 : Magnitude response of a complex input sinusoid for F(k) in a DFT filter bank where N=8, PRF=2 KHz and k=0 to N-1, kϵI showing all sidelobes

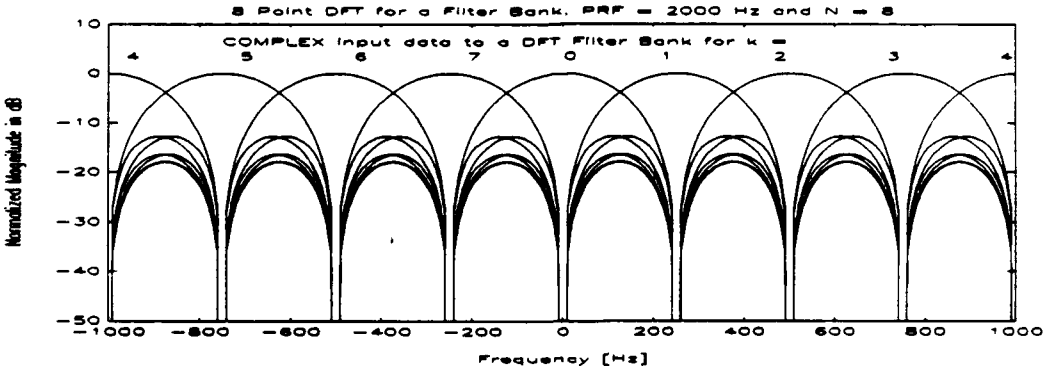


Figure 28 : Magnitude response in dB of a complex input sinusoid for F(k) in a DFT filter bank where N=8, PRF=2 KHz and k=0 to N-1, kϵI showing all sidelobes

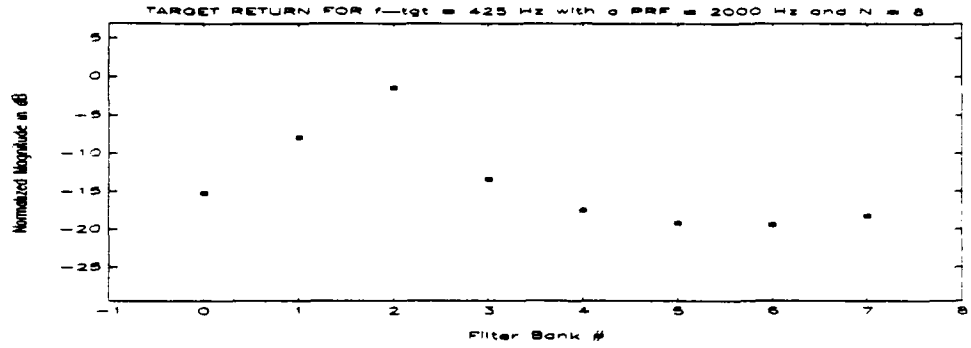


Figure 29 : Target response for each filter of a complex input sinusoid DFT filter bank where N=8, PRF=2 KHz and f_{tgt}=425 Hz

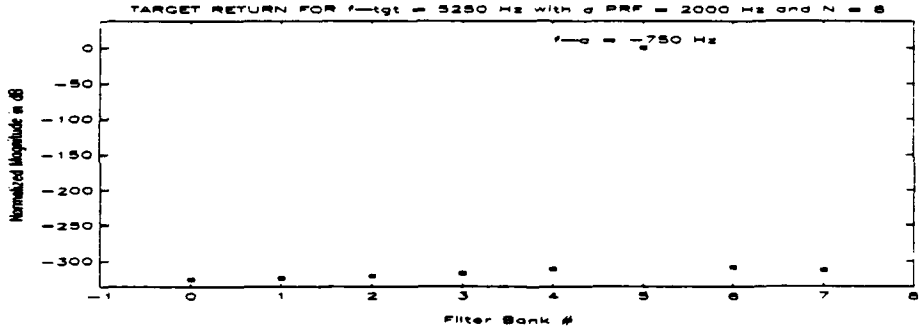


Figure 30 : Target response for each filter of a complex input sinusoid DFT filter bank where N=8, PRF=2 KHz and f_{tgt}=5250 Hz. Note that f_a=-750 Hz so anegative doppler target is incorrectly declared.

IV. WEIGHTED DFT FILTER BANKS

As seen in Chapters II and III the sidelobe response of a DFT filter is significant enough to cause problems in the detection of weak targets while maintaining a low false alarm rate. This problem results from the fact that a strong return from a single target could have enough of a magnitude response in the sidelobes of the other filters to exceed the detection threshold for those filters. Thus false target reports are generated, and the radar sees multiple targets when only one exists. If this problem is solved by raising the detection threshold to the point where the sidelobe response of a large target no longer causes false target declarations, then the radar has been desensitized to targets with smaller magnitude responses.

What is required is a method to lower the sidelobe response level of a DFT filter to an acceptable level. One way of doing this is through the use of windows (or weighting). This technique is in widespread use for communications systems and for other parts of radar signal processing such as reducing the time sidelobes in pulse compression. This chapter concentrates on some of the simpler DFT filter weighting schemes and describes the window functions that produce them. This chapter is limited to non adaptive weighting schemes in common use, such as:

- Rectangular window, where the inputs are uniformly weighted. The frequency domain transfer of the time domain weighting sequence is known as the Dirichlet kernel, $D_N(k)$.
- Bartlett (or Triangular) window where the inputs are weighted by a triangular function.
- Hanning (Von Hann, or raised cosine) window where the inputs are weighted by a raised cosine function such that the magnitude of the response at either end of the window is at zero.
- Hamming window which manipulates the Dirichlet kernel to further reduce the sidelobes immediately adjacent to the main lobe.
- Blackman window is an extension of the Hamming window in that it manipulates more terms of the Dirichlet kernel for greater sidelobe reduction. [Ref. 9:p566 to 574], [Ref. 14:p98 to 105]

A. WINDOWING

The DFT accepts an input sequence of N samples, s(n), and processes them according to

$$F(k) = \sum_{n=0}^{N-1} s(n)\, e^{-j\frac{2\pi nk}{N}} \tag{30}$$

A windowing function, w(n), weights each input sample by a specific amount. The amount each input sample is weighted is a function of the windowing scheme used. The new input sequence to the DFT is $s_w(n)$, and it is formed by

$$s_w(n) = w(n)s(n) \quad for\ n=1 \rightarrow N,\ n \in I. \tag{31}$$

The main benefit of the windowing scheme is a reduced sidelobe magnitude response; the chief disadvantage of a windowing scheme is that the mainlobe of the filter is

broadened. The penalty of mainlobe broadening due to windowing is that the filter width is increased; therefore, the ability to resolve targets and clutter close together in frequency is degraded. Also, there is a greater incidence of a single target being reported in adjacent filters if N is held constant.

The method to decrease the filter width to an acceptable level is by increasing N or by decreasing the PRF. Thus the search for better windowing functions has concentrated on finding windows that greatly reduce the sidelobe magnitude response, have as narrow a mainlobe as possible and are easy to compute and implement digitally.

B. WINDOWS USED IN THE SIMULATION

The windowing schemes that will be implemented in the simulation program DFTWIND.m at Chapter V are described below. Figure 31 is the input time domain weighting sequence for N = 50 for each of the windows in the following. [Ref. 9:p566 to 574]

1. Rectangular Window

The rectangular window is one where the inputs are uniformly weighted. This is the same as an unweighted window as developed in Chapter II and simulated in Chapter III. The input weighting sequence is therefore

$$w(n) = rect_N(n) \begin{cases} =1, & n=1 \rightarrow N,\ n \epsilon I \\ =0, & otherwise. \end{cases} \qquad (32)$$

The discrete frequency domain representation of this time domain weighting sequence is

$$D_N(k) = F(k) = \frac{\sin(\pi k)}{N \sin(\frac{\pi k}{N})}, \tag{33}$$

where $D_N(k)$ is known as the Dirichlet kernel.

2. Bartlett (or Triangular) Window

This is a weighting scheme in which the inputs are weighted by a triangular function. This allows for a smoother transition from zero to one for the input weights and reduces the sidelobes at the expense of broadening the mainlobe. The input weighting sequence is therefore

$$w(n) = (1-\frac{2|n|}{N}) rect_N(n). \tag{34}$$

The discrete frequency domain representation of this time domain weighting sequence is

$$F(k) = [D_{\frac{N}{2}}(\frac{k}{2})]^2. \tag{35}$$

3. Hanning (Von Hann, or Raised Cosine) Window

This is a weighting scheme in which the inputs are weighted by a raised cosine function such that the magnitude of the input at either end of the window is zero. This allows for a smoother transition from zero to one than for the Bartlett window so the sidelobes are reduced even more

However, this is at the expense of broadening the mainlobe again. The input weighting sequence is

$$w(n) = \frac{1}{2}[1+\cos(\frac{2\pi n}{N})]\,rect_N(n). \qquad (36)$$

The discrete frequency domain representation of this time domain weighting sequence is

$$F(k) = D_N(k) + \frac{1}{2}[D_N(k-1) + D_N(k+1)]. \qquad (37)$$

4. Hamming Window

This is a weighting scheme in which the inputs are weighted such that a zero will occur at the peak of the first frequency domain sidelobe. This splits the first sidelobe into two smaller sidelobes and is a very good method of reducing the appearance of a target report in adjacent sidelobes. This gain is paid for with an increased processing cost as N needs to be large for the best effects. Also, the mainlobe is broadened even further. However, the greatly reduced sidelobes may well be worth the cost. The input weighting sequence is

$$w(n) = [0.5435+0.4565\cos(\frac{2\pi n}{N})]\,rect_N(n). \qquad (38)$$

The discrete frequency domain representation of this time domain weighting sequence is

$$F(k) = D_N(k) + 0.41996\,[D_N(k-1) + D_N(k+1)]. \qquad (39)$$

5. Blackman Window

The Blackman window is an extension of the Hamming window in that it manipulates more terms of the Dirichlet kernel for greater sidelobe reduction. [Ref. 9:p573] Thus the sidelobes are reduced even more than under the Hamming window and the mainlobe broadened even more. This gain is paid for with an increased processing cost over the Hamming as more terms of the Dirichlet kernel must be used. Also, N needs to be large for the best effects. The input weighting sequence is

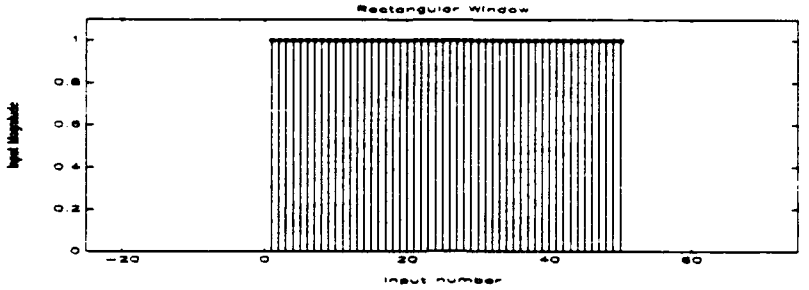
$$w(n) = [0.42+0.5\cos(\frac{2\pi n}{N}) +0.08\cos(\frac{4\pi n}{N})]\,rect_N(n). \quad (40)$$

The discrete frequency domain representation of this time domain weighting sequence is

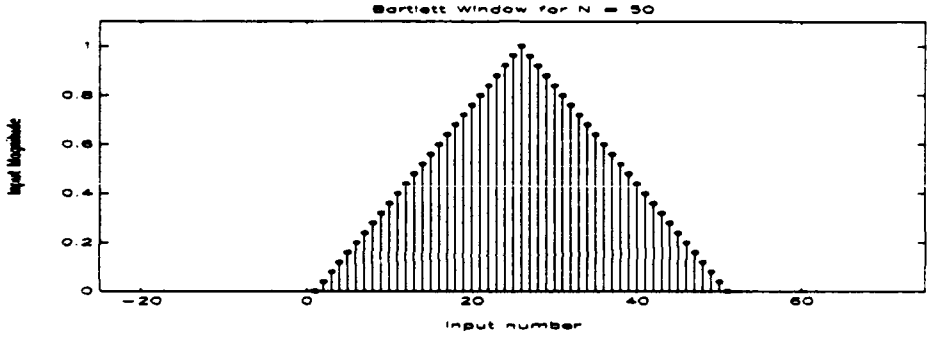
$$F(k) = D_N(k)+0.52083\,[D_N(k-1)+D_N(k+1)) +0.83333\,[D_N(k-2)+D_N(k+2)]. \quad (41)$$

C. COMPARISON OF WINDOWS USED IN THE SIMULATION

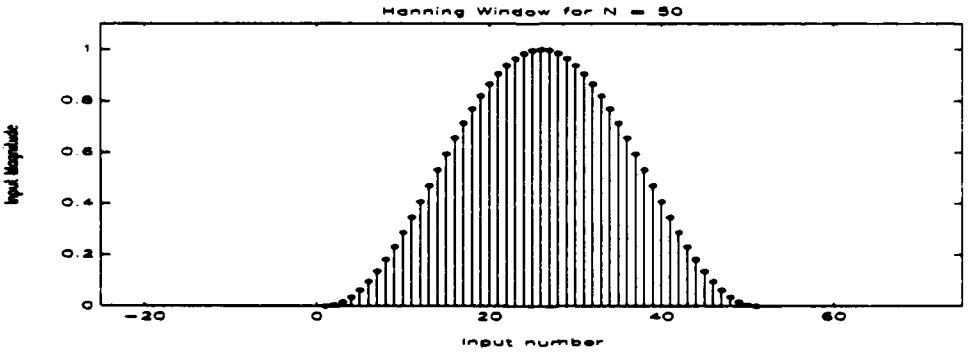
The following table is to be used as a rough guide when selecting which window function is likely to best suit the task. The processing cost is progressive as the windows are listed from least to most effective for reducing sidelobe



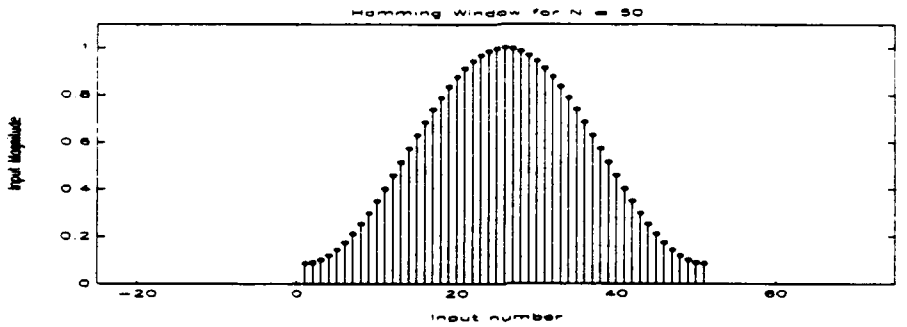
(a) Rectangular window



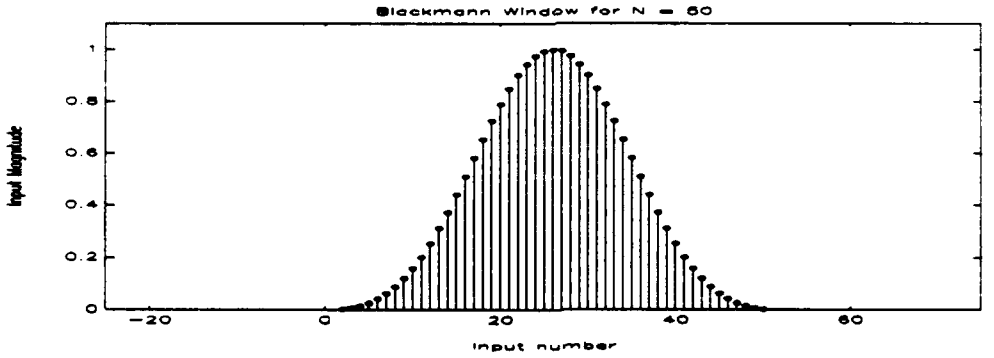
(b) Bartlett window



(c) Hanning window



(d) Hamming window



(e) Blackman window

Figure 31 : Time domain input weighting of the various window schemes that are to be simulated in Chapter V for N = 50.

levels. The increase to N required to compensate for the mainlobe broadening can be estimated as the mainlobes of the windows are given as being a per cent increase over that of a mainlobe for a rectangular window using the same N samples. Table 1 shows these results for the window schemes considered in this thesis.

TABLE 1 : CHARACTERISTICS OF VARIOUS WINDOWS

WINDOW	PEAK SIDELOBE LEVEL (in dB)	MAINLOBE BROADENING
Rectangular	- 13	0 %
Bartlett	- 27	44 %
Hanning	- 32	62 %
Hamming	- 43	46 %
Blackman	- 58	89 %

Other criteria not given in Table 1 that affect the choice of window scheme used are the rate at which the sidelobes fall off and the processing cost of the scheme. The rate at which the sidelobes fall off for each of the window schemes described above can be seen by using the simulation program at Chapter V. The processing cost can be estimated by noting how many iterations of the Dirichlet kernel need to be calculated for each scheme. Clearly the Hamming and Blackman windows require more processing time. It is worthwhile noting that the Hamming scheme gives both improved sidelobe reduction and a reduced mainlobe

broadening over the Hanning filter. This may well make the greater processing time required for the Hamming window more acceptable.

The simulation of these window schemes can be found at Chapter V, and the design process to achieve a suitable window scheme with the necessary sidelobe response and an acceptable filter width can be addressed.

V. SIMULATION OF A WEIGHTED DFT FILTER BANK

This chapter describes the methodology of the simulation of the response of a weighted DFT filter bank to a complex input sinusoid, looks at an overview of the program structure, describes how to use the program and shows how to interpret the results.

The weighted DFT filter bank simulation program will generates a filter bank showing the magnitude response to a complex sinusoidal input for all unambiguous frequencies. The following filter bank parameters can be varied in the simulation:

- N = the number of points required in the DFT
- PRF = the pulse repetition frequency of the radar
- k = the filter designator (if required)
- f_{tgt} = the doppler frequency of the target (if required).

A. PURPOSE OF THE SIMULATION PROGRAM

The equations implemented by the subprograms are for a quadrature sampled complex sinusoidal input. The frequency domain equations for the magnitude response of the windows to be processed were specified in Chapter IV. The increased processing time required by the computer for either the Hamming or Blackman windows will be evident.

B. PROGRAM STRUCTURE

The overall program structure can be seen in Figure 32. The structure is that of a main program (DFTWIND) which calls various subprograms (SINGWIN, WINBANK, ... etc) as required. There is an error message (ERRMSG) which detects an improper data entry, notifies the operator and restarts

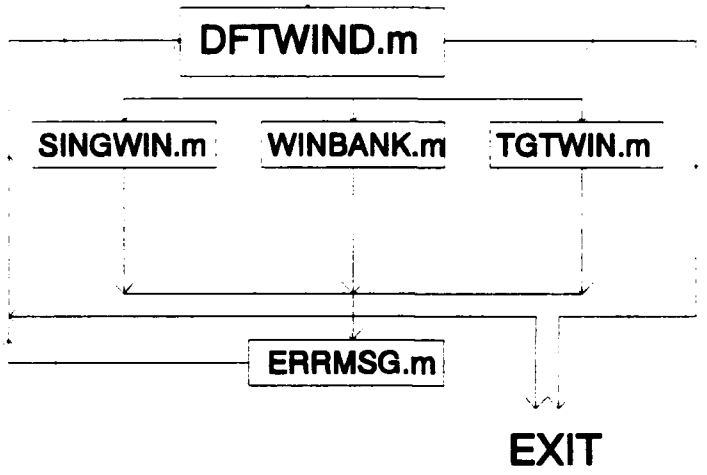


Figure 32 : Overall program structure for a weighted DFT filter bank simulation

the process. It is possible to exit the simulation from either the main or subprograms. This structure was used since it allows maximum operator control and input. It is a robust format that is data entry error tolerant. The other advantage of this format is that should an operator desire to change one of the subprograms or add another subprogram

the system can accommodate this. Any errors made in this process will affect only the subprogram in question. The rest of the simulation will remain operational while the new addition is being debugged.

C. OPERATING INSTRUCTIONS AND GUIDE

1. Initial Setup

To use this simulation the following MATLAB.m files must be copied into the user sub directory. Copies of these programs are available from Dr GS Gill at USNPGS.

- DFTWIND.m is the simulation control program.
- SINGWIN.m is the simulation subprogram responsible for displaying the magnitude response of a single filter across all unambiguous frequencies in the filter bank for the various windows.
- WINBANK.m is the simulation subprogram responsible for displaying the full magnitude response for all filters in the filter bank for all unambiguous frequencies in the filter bank for the various windows.
- TGTWIN.m is the simulation subprogram responsible for calculating and displaying the filter response for a target at a given doppler frequency. It can accept ambiguous target frequencies and the filter response will be for the alias frequency that lies in the filter bank. The filter that this alias frequency lies in can result in a positive doppler being identified as negative or vice versa.
- ERRMSG.m informs the operator when an entry error has been made and restarts DFTBANK after clearing the system of all variables.

This simulation will work on a DOS 286 machine, but a 386 or higher is preferable for increased speed. To start the simulation type DFTWIND [ENTER] at the MATLAB prompt.

2. Printing Graphical Outputs

The assumption has been made that the PRINT SCREEN key will be used to print out plots. To use this key the DOS graphics command must have been issued at the DOS prompt before any printing is attempted. C:\>> graphics [ENTER] is the necessary command.

Should it be preferable to create meta files for each plot the operator has two options. He can either

- Modify all plot options in the subprograms by adding a meta command and file name after the plot command, or
- Press the Ctrl-C keys to exit the simulation once the plot is on the screen. This will return him to the MATLAB prompt and he can issue the meta command from there. To restart the simulation type DFTWIND [ENTER].

3. Program Options

After the DFTWIND command has been issued to start the simulation then the first screen seen by the operator is at Figure 33. The second screen the operator will see is at Figure 34. At this point the operator must choose which type of system response that he wants to manipulate. Once this has been done the main program transfers control to the subprogram selected in Figure 35. The second screen is at Figure 36 and the third screen at Figure 37. All of

```
    This program will allow you to create weighted
Doppler Filter Banks for complex sinusoidal inputs.  The
window functions you can use to weight the input data
are;
      1.      RECTANGULAR WINDOW
      2.     BARTLETT (or Triangular) WINDOW
      3.     HANNING (Von Hann or raised cosine) WINDOW
      4.     HAMMING WINDOW
      5.     BLACKMAN WINDOW
   The options are to see the full response of a single
filter over the full spectrum of non aliasing
frequencies, to see the full response of a DFT bank of
filters or to see the magnitude response of each filter
in a DFT bank to a target occurring at a particular
frequency.
           TOUCH ANY KEY TO CONTINUE
```

Figure 33: First Screen of DFTWIND main control program

```
     You must now choose which type of response you wish
to see.
  To see a single filter response,          ENTER 1
  To see a weighted DFT filter bank,        ENTER 2
  To see target values for each filter,     ENTER 3

       TO EXIT THIS PROGRAM,ENTER 0
ENTER YOUR CHOICE NOW
CHOICE = 1
```

Figure 34: Second screen of DFTWIND main control program

```
IF YOU WANT A RECTANGULAR WINDOW,  CHOOSE 1
   IF YOU WANT A BARTLETT WINDOW,     CHOOSE 2
   IF YOU WANT A HANNING WINDOW,      CHOOSE 3
   IF YOU WANT A BLACKMAN WINDOW,     CHOOSE 4
   IF YOU WANT A HAMMING WINDOW,      CHOOSE 5

 Then you must decide if you want the response of the
window chosen to be plotted in comparison with a
rectangular window or by itself.
  The WINDOW Function you have chosen is # =
WINDOW = 3
 Compare with a Rectangular Window? Yes = 1, No = 2,
compare # = 2
```

Figure 35: First screen of a subprogram

```
  YOU MUST NOW ENTER SOME OF THE NECESSARY PARAMETERS.

  These parameters are the number 'N' of DFT points.
This is also the number of filters there will be in the
filter bank.
   Next you must choose the PRF you wish to consider.
   Then you must choose the 'k' offset for the DFT to
determine which filter in the DFT filter bank you will
select.  The choice of 'k' is an integer value usually
from 0 to N-1, but you are not limited to this selection
   Finally you must choose whether you wish the display
of the magnitude response to the DFT to be in normalized
dB or ratio form.

%  TOUCH ANY KEY TO CONTINUE
```

Figure 36: Second Screen of a subprogram

```
  Number of DFT points you wish is,  N =
  The PRF in Hz you wish is,  PRF =
  The Filter K # you wish to use is,  k =
  For a dB plot choose 1, for a ratio plot choose 2,  ? =
```

Figure 37: Third Screen of a subprogram input request

the screens for the subprograms are very similar, the only difference being whether "k" or "f_{tgt}" are required inputs.

First the choice of which window scheme to examine must be made. Then the decision whether or not to plot the selected window in comparison with a rectangular window or by itself must be made. The plots in the simulation are done in different colours and line styles so they are easily distinguished on the screen. Green is for the rectangular window and red for the window being studied. The colours will not show up on the printout of the plot so it can be difficult to differentiate between line styles. The choice of a dB or ratio plot on the third screen of a subprogram determines whether the magnitude scale on the y axis will be plotted for ratio values or in a decibel format. When attempting to determine the sidelobe level response the dB format is preferable. For a more general overview the ratio format is better.

Once all of the graphical outputs and results have been shown, each subprogram will terminate with the screen at Figure 38 where the operator can choose to do the same subprogram for the same type of input to the DFT (the SINUSOID variable remains constant), return to the main program DFTBANK to try a different simulation or exit the simulation.

Should any data entry error have occurred that the inherent MATLAB error routines did not recognize, than the simulation will detect it and the error message at Figure 39 will be displayed. All variables will be cleared and the main program DFTBANK will be restarted.

```
  YOU MUST NOW CHOOSE TO EITHER DO THIS CHOICE AGAIN,
RETURN TO THE MAIN MENU OR EXIT THE SIMULATION
   DO THE SINGLE LOBE AGAIN,          CHOOSE 1
   TO RETURN TO THE MAIN MENU,        CHOOSE 2
   TO EXIT THE SIMULATION,            CHOOSE 0
```

Figure 38: Final Screen of a subprogram

```
   YOUR CHOICE DOES NOT CORRESPOND TO THOSE AVAILABLE.
   THE PROGRAM WILL BE RESTARTED FOR YOU.
   REMEMBER YOU MUST CHOOSE 0 TO EXIT.

   PRESS ENTER TO CONTINUE...
```

Figure 39: Error message subprogram screen

D. INTERPRETATION OF RESULTS

The output of the simulation is mostly graphical in nature since graphical displays are readily assimilated and provide a more intuitive understanding of the process. Results from each of the subprograms will be presented. These plots will be analyzed to demonstrate some of the capabilities of the simulation program and to provide worked examples as a reference guide. In each case the required

data inputs will be provided and the selection process is described.

1. Single Filter Response

The SINGWIN subprogram displays the magnitude response of a single DFT filter weighted according to choice across all unambiguous frequencies in the filter bank. These unambiguous frequencies are for $|f| \leq PRF/2$. While it is the frequency range of the mainlobe magnitude response of the filter that defines the frequency coverage of the filter, however there can be a response to any frequency, even from the sidelobes of the filter. If the response for a frequency were from the sidelobes of the filter and it exceeded the detection threshold set for that filter, a false target report for that filter would be given.

The operator can vary the PRF and N in the simulation to achieve a filter response with an acceptable

- filter width,
- frequency resolution, and
- doppler frequency coverage.

The data at Table 2 is good for a quick comparison of the relative windows and can help in determining the most likely windowing scheme to satisfy a radar design requirement. Table 2 shows these results for the windowing schemes that were addressed in this thesis.

TABLE 2 : WINDOW CHARACTERISTICS

WINDOW	PEAK SIDELOBE LEVEL (in dB)	MAINLOBE BROADENING
Rectangular	- 13	0 %
Bartlett	- 27	44 %
Hanning	- 32	62 %
Hamming	- 43	46 %
Blackman	- 58	89 %

In the following example the inputs to the simulation were

- CHOICE = 1 for a single filter response
- WINDOW = 2 for a Bartlett window
- Compare = 2 so it will not be compared to a rectangular window
- N = 8
- PRF = 2000 Hz
- k = 0 for the F(k) magnitude response
- dB_ratio = 2 for a ratio plot

The plot resulting from these simulation choices is at Figure 27. Had dB_ratio = 1 been selected than the dB ratio magnitude response at Figure 28 would have been plotted.

2. Weighted DFT Filter Response

The WINBANK subprogram displays the mainlobe magnitude response of all filters in the filter bank for all unambiguous frequencies. The choices are the same as for

SINGLOBE, with the exception of "k" since the mainlobe magnitude responses for k=0 to N-1, k ϵ I are all plotted automatically.

In the following example the inputs to the simulation were

- CHOICE = 2 for a full filter bank response
- WINDOW = 4 for a Hamming window
- Compare = 1 so it will be plotted versus a rectangular window
- N = 16
- PRF = 2000 Hz
- dB_ratio = 1 for a dB plot

The plot resulting from these simulation choices is at Figure 41. From it the filter width at crossover of the mainlobes, straddle loss and other performance measurements in comparison to an equivalent rectangular window DFT can be noted. This demonstrates where the detection threshold can be set for the weighted window in comparison to the unweighted rectangular window if it is desired to

- avoid declaring targets in more than one mainlobe of a filter, then the threshold must be set higher than the mainlobe crossover point, or
- avoid declaring targets from the sidelobe of a filter, then the threshold must be set higher than the highest sidelobe.

This gives the operator a good feel for the change in dynamic range for the different windows over which he can

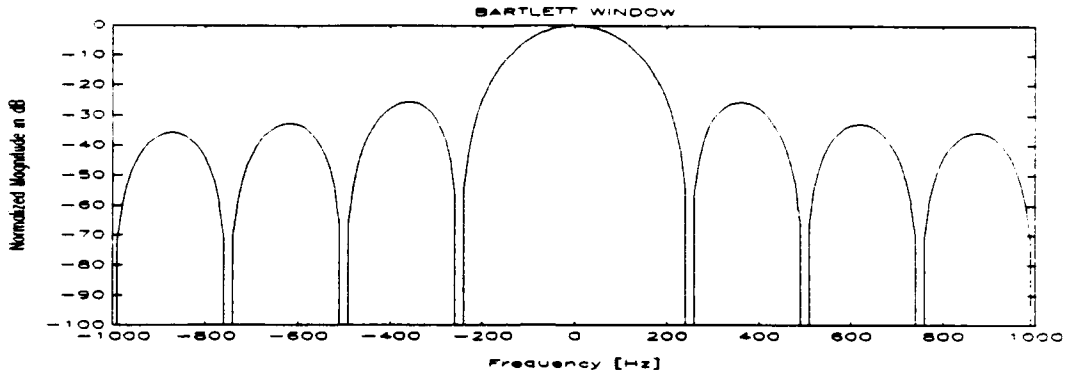


Figure 40 : Magnitude response for the F(0) filter of a Bartlett weighted DFT

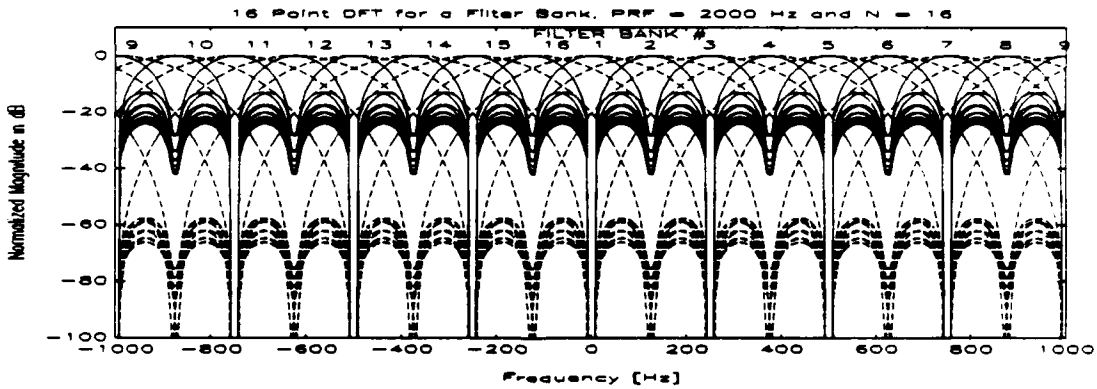


Figure 41 : Magnitude response for the full filter bank using a Hamming weighted DFT in comparison to an unweighted rectangular DFT

detect targets that vary in signal strength and not affect his false alarm rate.

The graph is difficult to analyze when it is printed in black and white, especially if the additional information from the rectangular window for reference purposes is on it as well. The graph is much more intelligible when viewed on screen as the different colours make interpretation easier.

3. Filter Bank Response to a Specific Target Frequency

The TGTWIN subprogram is the same as the WINBANK subprogram, except that it will require an additional input for the doppler frequency the target or clutter is to appear at and that all plots are automatically of a dB nature. If the doppler frequency given does not satisfy the Nyquist criteria then the alias frequency (f_a) will be processed by the filter bank. The filter that the alias frequency lies in has no direct relation as to whether the original doppler was a positive or negative doppler frequency. Thus, this can result in a positive doppler being identified as negative or vice versa.

In the following example the inputs to the simulation were

- CHOICE = 3 for a target report,
- WINDOW = 4 for a Hamming window,
- Compare = 1 so it will be plotted versus a rectangular window,
- N = 16,

- PRF = 2000 Hz,
- dB_ratio = 1 for a dB plot, and
- f_{tgt} = 356 Hz.

The plot resulting from these simulation choices is at Figure 30. The sidelobe improvement of the Hamming window

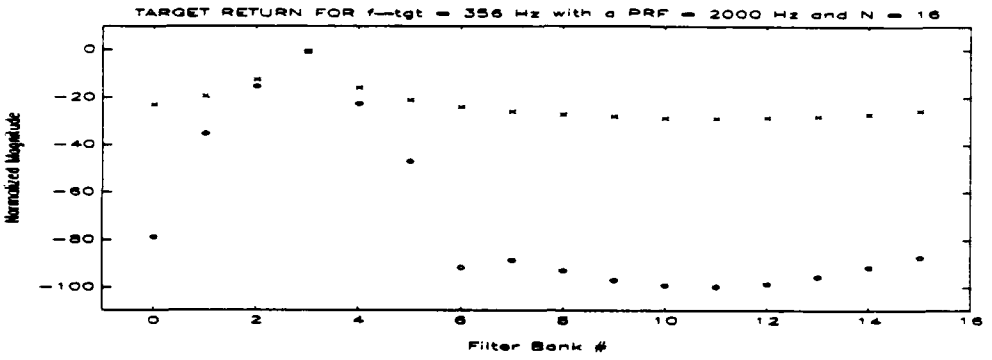


Figure 42 : Target report for a Hamming window (o) in comparison to that from a Rectangular window (x). The Hamming window shows the steep sidelobe drop off that makes it useful

over that of an unweighted DFT is evident, though it is worth noting that the sidelobes adjacent to the mainlobe do not see the same improvement that the other sidelobes do.

VI. PULSE COMPRESSION TECHNIQUES

Pulse compression (PC) is a technique in which a long pulse that has been modulated in either frequency, amplitude or phase is transmitted. The received signal is then processed in a matched filter to compress it to a narrow pulse. Doing so has the following advantages and disadvantages [Ref. 6:p420 to 434], [Ref. 4:p123 to 156] and [Ref. 11]:

- A long pulse can be used to increase the average transmitted power.
- The short pulse in use after compression is desired for range resolution, accurate range measurements and discrimination of multiple targets.
- The radar is less vulnerable to ECM that cannot generate the same coherent waveforms as the radar.
- The target signal experiences a processing gain while the noise in the system does not.
- Some pulse compression waveforms are sensitive to doppler effects and this may degrade the overall performance of pulse compression.
- The minimum range at which the target can be detected is increased for a radar with pulse compression.

A. RADAR RANGE AND RESOLUTION

Two of the key characteristics of a radar system are a long detection range and a narrow range resolution (ΔR). The requirement for a long detection range is self evident. The reason to have a narrow range resolution is to resolve

two targets that are close together. Both of these characteristics are linked by a dependence on the pulse length of the transmitted waveform.

1. Range

The range dependence on the pulse width (T) can be seen from the radar range equation as follows:

$$R^4 = \frac{P_{AVG} G^2 \lambda^2 \sigma}{(4\pi)^3 F_n k T_0 B L_s}, \tag{42}$$

for

$$P_{AVG} = \frac{P_{PEAK} * T}{PRI}, \tag{43}$$

where

- R = the radar detection range,
- P_{AVG} = the average power of the transmitted pulse,
- G = the gain of the monostatic radar's antennae,
- λ = the wavelength of the transmitted waveform,
- σ = the Radar Cross Section (RCS) of the target,
- F_n = noise figure,
- k = Boltzman's constant,
- T_0 = the temperature the radar is operating at,
- B = the filter bandwidth,
- L_s = the system losses in the radar,
- P_{PEAK} = the peak power the radar is operating at,
- T = the pulse width of the transmitted waveform, and

- PRI = the pulse repetition interval.

Thus if all other factors in Equation 42 are held constant except for P_{AVG}, the only way to increase the detection range is by increasing either P_{PEAK} or T. There are practical limits on the peak power a radar transmitter can generate. With all other factors in Equation 35 remaining constant, the only way to effect a further increase in the range of the radar is by increasing T.

2. Range Resolution

A good range resolution (ΔR) is desirable for the following reasons: [Ref. 6:p421]

- Better range resolution leads to better range accuracy.
- When attempting to separate multiple targets it is necessary to separate them in either range, azimuth or elevation. Given the 3 dB beamwidth of antennas suitable for search radar applications a good angle resolution in either azimuth or elevation is difficult to achieve. Thus it is preferable to separate multiple targets in range rather than in angle. Therefore the smaller the range resolution is, the better the ability to separate multiple targets in range will be.
- The amount of clutter a target has to compete against varies directly with ΔR. Thus the smaller ΔR is, the less clutter there is to raise the CFAR detection level. Therefore it will be possible to detect smaller magnitude targets.

The equation for the calculation of a radar's ΔR is as follows:

$$\Delta R = \frac{c\,T}{2}. \qquad (44)$$

Clearly ΔR depends only on T, hence a shorter T is desirable. Therefore there is a conflict between the requirements for a high range resolution (small T) and a large detection range (large T). One way to try to meet both of these requirements is through pulse compression.

B. PULSE COMPRESSION

Pulse compression (PC) is a scheme to resolve the conflicting requirements of narrow ΔR and long detection range. In pulse compression a long pulse (T) composed of several sub pulses (τ) which have been modulated in phase, frequency or amplitude is transmitted. The received signal is correlated with the replica of the transmitted pulse. This results in the correlation having a high impulse like output when there is a return from a target and a low output in the absence of a target. The long transmitted pulse, consisting of all the sub-pulses, increases P_{avg} to its desired value. Thus P_{avg} continues to be dependent on T. However, the narrow pulse output of the correlator will satisfies the ΔR requirement and is not dependent on T.

1. Pulse Compression Ratio

The relationship amoung T, τ and the PRI is seen in Figure 43. If the transmitted pulse width is T and the subpulse width is τ, then,

$$T = N\tau . \tag{45}$$

The pulse compression ratio (PCR) is

$$PCR = \frac{T}{\tau} . \tag{46}$$

The pulse compression achieved in a real world system is usually some value less than or equal to the PCR. Any losses due to doppler or other effects are referred to as the pulse compression loss factor.

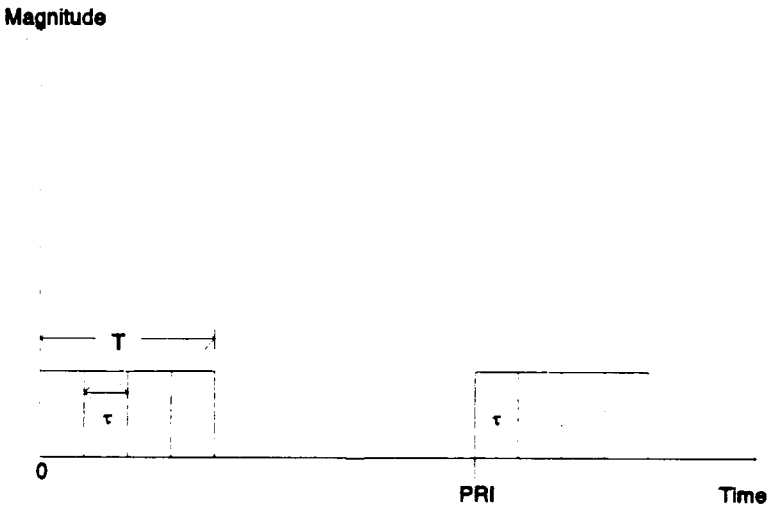


Figure 43 : Relationship between the pulse width (T), the subpulse width (τ) and the PRI for pulse compression

2. Range Resolution

The range resolution under pulse compression is a function of the subpulse length. Therefore, for pulse compression coding schemes

$$\Delta R = \frac{c\tau}{2}.$$

Now $P_{AVG} \propto T$ while $\Delta R \propto \tau$; thus with pulse compression, it is possible to achieve both a good detection range and a good range resolution.

3. Pulse Compression Processing Gain

Processing gain is the amount by which the SNR at the output of PC improves with reference to the SNR at the input. Noise at the input is random and has a low correlation with the replica of the transmitted pulse. However, the target return has a high correlation and receives a processing gain equal to the achieved pulse compression. Therefore, the SNR is improved by a factor equal to the pulse compression. This is one of the significant advantages of using pulse compression techniques.

4. Eclipsing Loss

When a radar pulse is being transmitted, it is not possible for the receiver to operate. Any target information that appears during this time interval is lost. For a radar using pulse compression, the length of the transmitted pulse is increased and more eclipsing loss will result. This will affect the minimum range (R_{min}) at which the radar can detect targets. Also, if the radar is designed to detect targets

beyond the unambiguous range (R_u), the eclipsing loss is repeated at each multiple of R_u.

C. PULSE COMPRESSION PERFORMANCE CRITERION

The measures of PC performance used in the simulation programs at Chapter VII are defined as follows;

1. Peak to Sidelobe Ratio

The peak to sidelobe ratio (PSL) is the ratio of the highest sidelobe to the mainlobe. High sidelobes tend to cause false alarms when they cross the detection threshold setting. To reduce the false alarm rate, the detection threshold must then be raised, thereby desensitizing the radar. It can be seen that a small target could thus be missed by the radar.

2. Integrated Sidelobe Ratio

The integrated sidelobe ratio (ISL) references the effect of the sum of all the sidelobes to the mainlobe. It is desirable to have the maximum energy possible in the mainlobe, where it will be used for detection purposes since any energy in the sidelobes is discarded insofar as target detection is concerned.

3. Pulse Compression Ratio

The pulse compression ratio (PCR) is the ratio of the transmitted pulse width to the compressed pulse width.

4. Doppler Sensitivity

When the transmitted radar pulse encounters a moving target, a doppler shift occurs in the reflected pulse. A doppler shifted waveform from a moving target may not compress to its full potential and thus provides only a limited processing gain. This effect varies from code to code. Codes which can continue to compress effectively in spite of doppler shifts are known as doppler tolerant codes and thus are preferable.

Target motion causes a shift in frequency given by;

$$f_d = \frac{2\ V_d}{\lambda}. \qquad (48)$$

The corresponding change in phase is given by:

$$\phi_d = 2\pi f_d\, n\tau \quad for\ n=1\text{-}N,\ n \in I, \qquad (49)$$

where

- V_d = the relative doppler velocity between the target and the radar,
- f_d = the doppler frequency resulting from V_d,
- λ = the wavelength of the transmitted pulse,
- ϕ_d = the doppler phase shift resulting from V_d,
- n = the subpulse number, and
- τ = the subpulse width.

Since the doppler phase shift is progressive from subpulse to subpulse, this may pose a serious problem to phase coded pulse compression schemes. It can cause

reductions in the processing gain and increases in the sidelobes. If a code is not doppler tolerant then there are schemes available to attempt to reduce the influence of the doppler effect. One of these schemes is described in the simulation of pulse compression in Chapter VII.

D. TYPES OF PULSE COMPRESSION

The three basic types of modulation schemes in use are amplitude, frequency and phase modulation.

1. Amplitude Modulation

This modulation scheme is almost never used since the amplitude modulation of a pulse train is so easily distorted by numerous of external factors. [Ref. 6:p420]

2. Frequency Modulation

Historically the most common pulse compression schemes used have been those of the frequency modulated form. In this scheme the frequency modulation of the carrier is varied throughout the overall pulse length. This variation can be either linear or non-linear and it can be implemented in either a continuous or discrete form. Therefore the four subsets of frequency modulation are; [Ref. 3:p10.4], [Ref. 2:p9 to 25] continuous linear frequency modulation, discrete linear frequency modulation, continuous non linear frequency modulation and discrete non linear frequency modulation.

a. Continuous Linear Frequency Modulation

The most common type of this form of modulation is linear frequency modulation (LFM). In LFM, there are no actual subpulses since the waveform is continuous. The frequency of each transmitted pulse of duration T is increased or decreased at a constant rate over the pulse length as shown in Figure 44. The matched filter for this waveform has a "lagtime versus frequency" characteristic that is the opposite of the transmitted pulse. Thus when the returned signal is passed through the matched filter, the trailing portions of the signal "catch up" to the

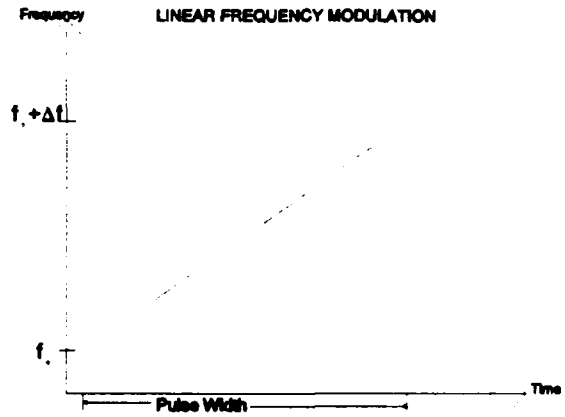


Figure 44 : Linear frequency modulation scheme

leading edge, and a compressed output with a greater amplitude and smaller pulse width results. [Ref. 1:p217] The amplitude gain over the transmitted signal is the PCR, and the compressed pulse width corresponds to τ=T/PCR. The

design specifications for LFM are as follows; [Ref. 2:p13], [Ref. 12]

- Δf is the minimum resolvable frequency difference, and it is defined as being $\Delta f = \tau^{-1}$.
- In order to achieve Δf the modulation rate (F) over the pulse length must be at a rate no less than $F = \Delta f / T$.
- Clearly for a fixed T, the width of the compressed pulse is determined by the modulation rate. The greater the rate of change for f as seen in Figure 45, then the smaller τ will be.
- The overall pulse is formed by starting with a pulse equal to τ. Then the frequency is linearly changed at a rate where $T = \tau + k\Delta f$. Here k is the dispersion coefficient in units of sec/Hz/sec from the dispersive delay line that forms the pulse T. Since in almost all cases $k\Delta f \gg \tau$, the PCR is considered to be

$$PCR = \frac{T}{\tau} = \frac{\tau + k\Delta f}{\tau} \approx k\Delta f. \qquad (50)$$

Target doppler causes an error in the range measurement for the LFM waveform. However, there is no other significant adverse effect due to the doppler shift. Range errors can be corrected if the target doppler shift is known.

The other problem of LFM is that the time sidelobes of the compressed waveform are large enough to affect radar sensitivity and the constant false alarm rate (CFAR). Large targets have significantly large sidelobes which interfere with the detection of small targets. If the detection threshold is set to detect small targets, then the sidelobes of large targets are reported as targets as well.

If the threshold is raised high enough to avoid this problem, than small targets go undetected.

The chief advantages of the LFM waveform are its doppler tolerance and the ability to achieve very large

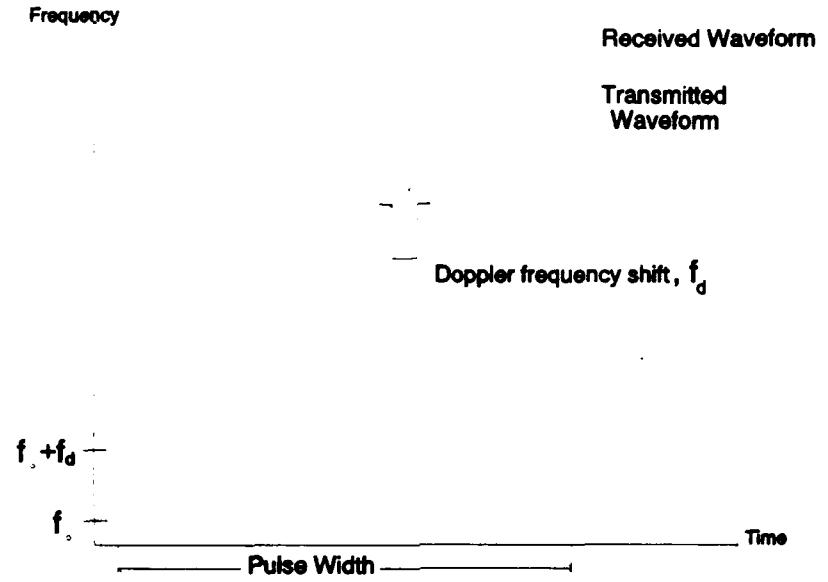


Figure 45 : Linear frequency modulation scheme undergoing a doppler frequency shift

PCRs. The time sidelobe problem can be reduced through the use of weighting schemes, such as those introduced to reduce the frequency sidelobes of the DFT filter bank in Chapter IV.

b. Discrete Linear Frequency Modulation

In discrete linear frequency modulation the frequency is constant over a given subpulse interval τ and then instantaneously changed by an amount Δf for the next subpulse, unlike the continuous linear change for LFM as

seen in Figure 46. This is done for a total of N subpulses. The best frequency step interval found has been to have $\Delta f=\tau^{-1}$ which avoids undesirable pulse compression waveform sidelobes. Digital LFM has the same characteristics as LFM. [Ref. 2:p13]

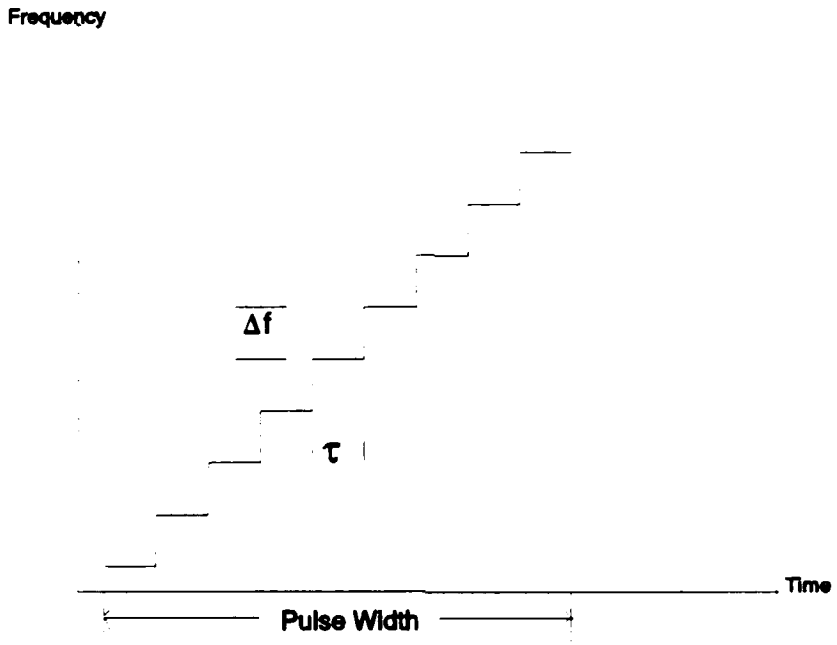


Figure 46 : Digital linear frequency modulation scheme

c. Continuous Nonlinear Frequency Modulation

Nonlinear FM waveforms can be used in modern radar systems. They have some advantages over linear FM waveforms, the most important being their greatly reduced time sidelobes after compression. However, their chief disadvantage of expensive hardware implementation outweighs this gain, except for very large frequency invariant

systems. This waveform is used in FAA radars for enroute commercial aircraft surveillance.

d. Discrete Nonlinear Frequency Modulation

Digital waveforms are easier to implement and process in modern computers. However, it is doppler intolerant and just as expensive to implement as the continuous nonlinear FM.

3. Phase Modulation

This is another form of pulse compression modulation in practical use. Its advantages include better time sidelobes than for FM, the ease of generating the waveform and use of the same hardware to generate different length codes. There are two basic types of phase modulation in use today, and they are binary phase or polyphase modulation. [Ref. 3:p10.15 to 10.27], [Ref. 2:p17], [Ref. 12]

a. Binary Phase Modulation

In binary phase modulation, the phase changes to the waveform are in either 0 or π increments from subpulse to subpulse for a total of N subpulses. Since $T=N\tau$, then by definition the PCR is

$$PCR = \frac{T}{\tau} = N. \qquad (51)$$

The time sidelobes from the correlator for binary codes need to be minimized as much as possible, just as in the FM case. Thus, based on this criterion, the binary phase modulation

schemes are broken into two subclasses, namely Barker and pseudo random codes.

(1) Barker Codes. The Barker codes are known as perfect codes because their time sidelobes are all equal and of unit magnitude. Thus the waveform has the minimum possible energy contained in the ISL, and all sidelobes are at the same PSL. [Ref. 2:p17] However, they are only known to exist up to a maximum length of 13. After this length, it is possible to use combined Barker codes for a greater PCR in which the Barker subpulses are themselves coded within another Barker code. While this extends the PCR, the PSL remains the same as that of the lowest order Barker sequence used in the combined coding, and the ISL worsens. The main problem with Barker and combined Barker codes is their low tolerance to doppler shifts as the performance of the received waveform in the matched filter is seriously affected; PCR, PSL and ISL suffer. If the transmitted pulse length is short, the adverse doppler effects are kept within tolerable limits for the Barker codes. However, when the combined Barker is used to increase the PCR, its greater length leaves it vulnerable to the adverse doppler shift effects. The Barker codes are not particularly immune to the effects of ECM. Once the waveform is detected and identified it will be susceptible to ECM that has the capability to generate coherent phase coded waveforms.

(2) Pseudo Random Codes. If the binary sequence transmitted is a purely random sequence of 180° phase shifts for N subpulses, then the PCR is still be equal to N. However, the PSL and ISL characteristics of the waveform are most likely poor due to high time sidelobes resulting from the correlation process. For this reason, truly random codes are seldom used. However, an attempt to approximate random phase codes whilst achieving the best ISL and PSL possible for that length of code is made because of the high ESM tolerance of random codes. The most common form of the pseudo random codes is the one known as maximal length sequence, where an "n" stage shift register is used with feedback to generate the code. The code will be of length $N = 2^n-1$, and the code created by this process depends on the "n" initial conditions set in the shift registers and the form of the feedback loop. The feedback loop can be a modulo adder, XOR gate or any other logical device. Figure 47 illustrates this process for a modulo 2 adder with inputs from registers three and four for a four stage register. [Ref. 3:p10.18 to 10.20], [Ref. 6:p428 to 431]

Clearly a great number of codes can be generated for any given N. Many of these codes have poor ISL and PSL performance on compression and are discarded. The code or codes which for any given N show the best ISL and PSL performance are known as the optimal codes. A great

many optimal codes are known to exist and are in use today. For example, for N=40, there are 114 known optimal codes, and thus the choices for ESM suppression are widespread. [Ref. 3:p10.18]

The same problems that occur with binary codes in respect of doppler phase tolerance also occur with pseudo random codes. In addition, since the time sidelobes of pseudo random codes are not of unitary height the doppler performance of pseudo random codes is usually worse than for the Barker codes.

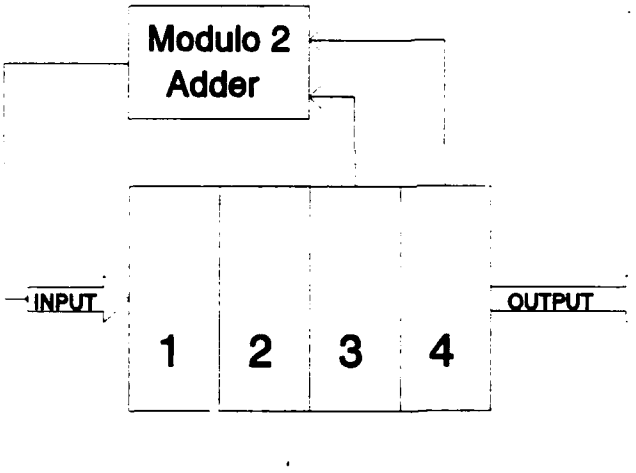


Figure 47 : A pseudo random code generator using an adder with four initial conditions to be set. A maximal length 15 bit pseudo random code can be generated.

b. Polyphase Modulation

Rather than having the subpulses use phases of 0 or 180°, the various polyphase schemes use multiple values

between 0 and 360° for the subpulse phase modulation. The schemes for the creation of the various polyphase schemes vary significantly in scope, characteristics and results. Those polyphase codes based on nonlinear waveform generation have the same problems as do the nonlinear FM codings discussed earlier. For this thesis, only some of the polyphase schemes based on the LFM modulation scheme as adapted for polyphase implementation are examined. These codes are the Frank polyphase code and the P3 polyphase code. Since both of these codes are derived from the LFM waveform, they possess all of the properties, limitations and advantages of LFM modulation. Since they can be implemented digitally, they are easy to use in modern radars. [Ref. 3:p10.25], [Ref. 2:p15]

VII. SIMULATION OF PULSE COMPRESSION

This chapter describes the methodology of the simulation of pulse compression for a variety of modulation schemes and examines one of the methods of correcting for doppler shifts in pulse compression. It also looks at an overview of the program structure, describes how to use the program and shows how to interpret the results.

The pulse compression simulation program will generate the output of the correlator for the chosen modulation scheme taking into account any doppler effects present. The shape of the correlation is displayed graphically, along with the PCR, PSL, ISL and the pulse compression achieved by the correlator. The following parameters can be varied in the simulation:

- The type of modulation scheme to be used,
- N = the number of subpulses in the modulation scheme,
- PRF = the pulse repetition frequency (if required),
- RF = the frequency the radar is transmitting at (if required),
- τ = tau, the subpulse width,
- V_{tgt} = the actual speed of the target (if required), and
- V_{max} = the assumed maximum speed of the target (if required).

The modulation schemes that are simulated are:

- Barker codes from binary phase modulation,

- Combined Barker codes from binary phase modulation,
- Pseudo Random codes with a modulo 2 XOR gate from binary phase modulation,
- P3 codes from polyphase modulation (which has LFM characteristics), and
- Frank codes from polyphase modulation (which has LFM characteristics).

By controlling these parameters the student can address the realities of some of the pulse compression schemes in current use. Some of the situations that can be addressed are:

- The various performance criteria of a given modulation scheme can be assessed.
- The performance criteria of a given modulation scheme with a target doppler shift can be assessed. The doppler shift can either be for a given target velocity or an assumed doppler shift across the specified pulse width.
- The effects that the transmitted radar waveform pulse width will have on the eclipsing loss for a specific radar can be assessed. This is presented graphically by displaying both T and the PRI over three pulses. The minimum range (R_{min}) that the radar can detect at and its unambiguous range (R_u) will be given.
- There are numerous other possibilities left to the user's initiative.

A. PURPOSE OF THE SIMULATION PROGRAM

The modulation schemes implemented by the sub programs are detailed below.

1. Barker Code

The Barker codes are the only codes which have sidelobes of unit magnitude and thus yield the best results

for their lengths. They are formed by changing the phase of the transmitted frequency by a phase shift of 0 or π for each τ interval. These 0 and π phase shifts are normally expressed by taking their cosines and thus representing them as binary codes of ± 1. There are N subpulses and the PCR is

$$PCR = \frac{T}{\tau} = N. \qquad (52)$$

Although this code is doppler intolerant its short length may help shield it from the worst effects. The known Barker codes are listed in Table 3. An example of the output of the correlator for a Barker 13 code without doppler effects is at Figure 48.

2. Combined Barker Code

This is where a Barker code is coded within another Barker code to increase the length of the code, and thus the PCR. PSL remains the same as that of the lowest Barker code used, and the ISL usually worsens. This code is not doppler tolerant, and since it is longer than the Barker codes it is more susceptible to doppler. It is usually referred to as a 'm' in 'n' combined Barker code, where m is the inner code and n is the outer code. Thus there are a total of N=m*n subpulses, and the PCR is

TABLE 3 : THE KNOWN BARKER CODES

Barker Code	Binary Format	PSL [dB]	ISL [dB]	PCR
2	1,-1 or 1,1	-6.0	-3.0-	2
3	1,1,-1	-9.5	-6.5	3
4	1,1,-1,1 or 1,1,1,-1	-12.0	-6.0	4
5	1,1,1,-1,1	-14.0	-8.0	5
7	1,1,1,-1,-1,1,-1	-16.9	-9.1	7
11	1,1,1,-1,-1, -1,1,-1,-1,1,-1	-20.8	-10.8	11
13	1,1,1,1,1,-1, -1,1,1,-1,1,-1,1	-22.3	-11.5	13

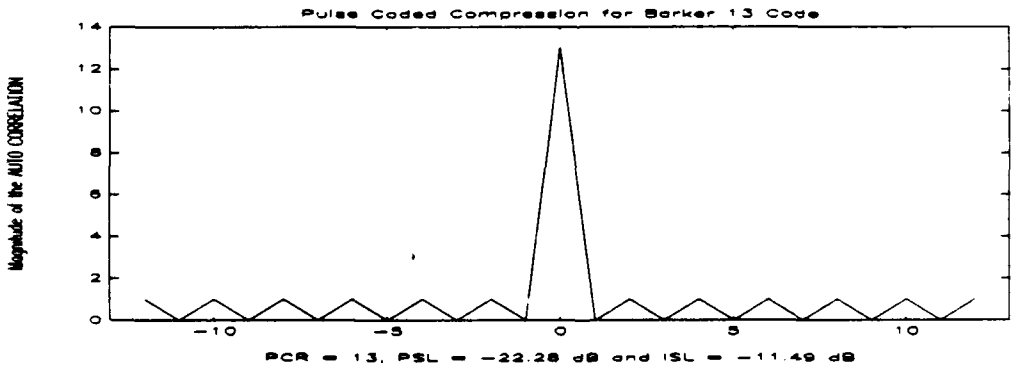


Figure 48 : Output of the correlator for a Barker 13 code without doppler effects

$$PCR = m*n = N. \quad (53)$$

An example of a 13 in 13 combined Barker code is at Figure 49.

3. Pseudo Random Codes

The pseudo random codes can be generated for a length n shift register using a modulo 2 XOR gate. Any two of the shift register cells can be selected to be fed to the XOR gate. The shift registers to be used are referred to as PN(3,4) when the 3rd and 4th shift register cells are to be used as feedback to the XOR gate. The number of "n" initial conditions must also be specified. The maximum length for a unique code is $N=2^n-1$. This code is useful for ECM purposes, but has a poor response to uncorrected doppler conditions. Also, it has large sidelobes when compared to the Barker codes, which is a major disadvantage. The correlation of a n=4, PN(3,4) code with all initial conditions set to 1 is at Figure 50.

4. P3 Polyphase Code

There are many variations of this type of code, but the one considered here is the P3 code derived from LFM (which makes it doppler tolerant). The mathematical form for the generation of the code phases for the P3 polyphase code is

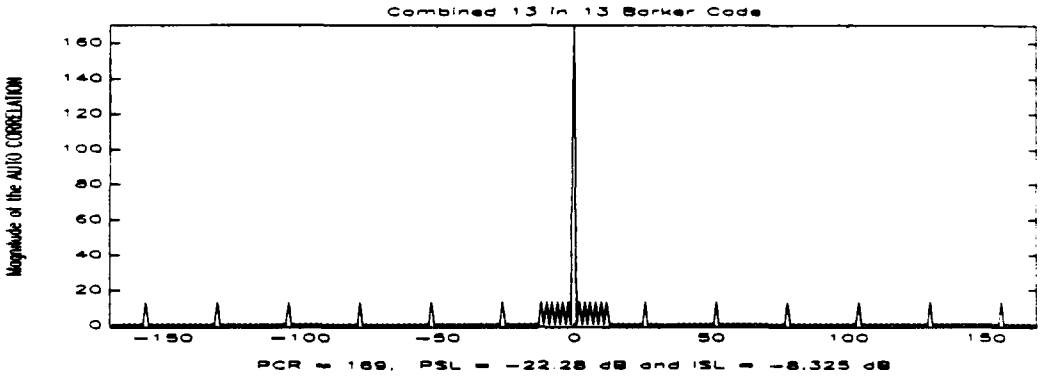


Figure 49 : Output of the correlator for a combined Barker 13 in 13 code without doppler effects

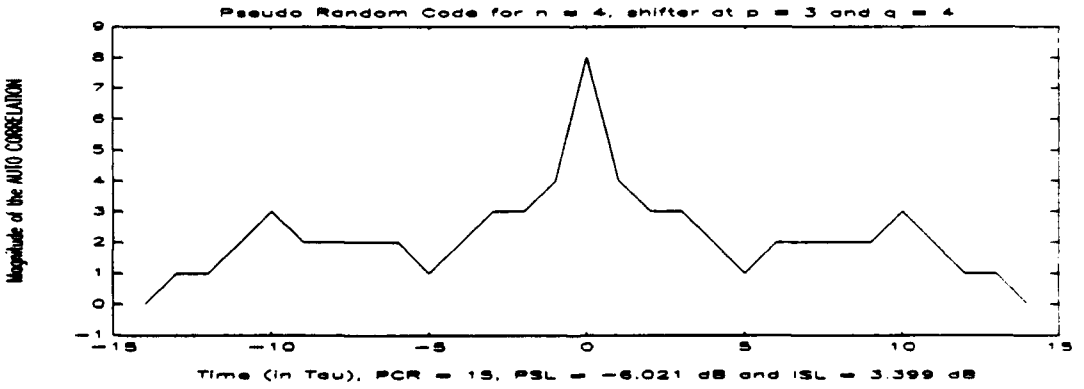


Figure 50: Output of the correlator for a pseudo random code without doppler effects for 4 initial conditions (all equal to 1)and an XOR gate feedback from bins 3,4

$$\phi(i) = \frac{\pi}{N}(i-1)^2 , \quad i=1-N, \quad i \epsilon I, \tag{54}$$

where N is the number of subpulses desired (also the PCR). An example of the output of the correlator for the P3 code is at Figure 51.

5. Frank Code

The mathematical form for the generation of Frank code phases is

$$\phi(i,j) = \frac{2\pi}{N}(i-1)(j-1) \tag{55}$$

where i=j=n, the Frank code number, n ϵ I, and N = n^2 is the number of subpulses present.

The code is formed in much the same fashion as the combined Barker where i corresponds to the inner code and j to the outer code. The resulting ϕ matrix is reshaped into a vector subpulse train. An example of the output of the correlator for a Frank code with n=4 and N=16 is at Figure 52.

B. PROGRAM STRUCTURE

The overall program structure can be seen in Figure 53. The main program (PCRSIM) which calls one of three subprograms NODOPP, DOPP or XDOPP depending on whether processing is done without the effects of doppler, with

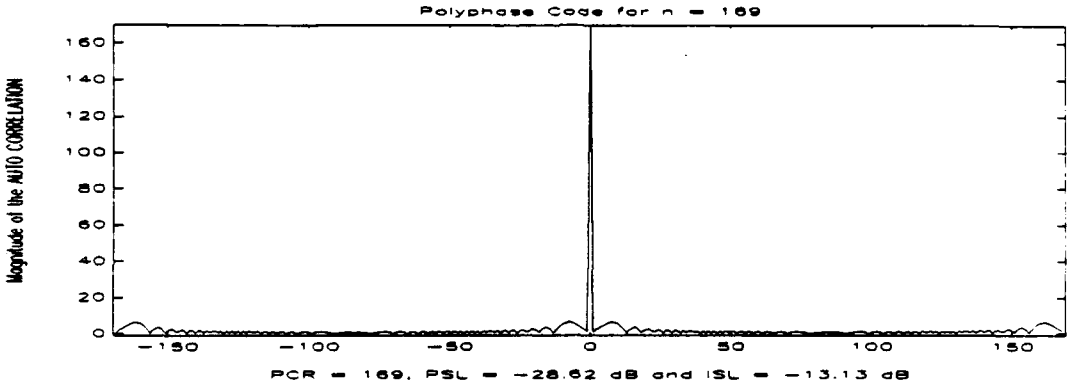


Figure 51: Output of the correlator for a P3 polyphase code for N = 169 without doppler effects

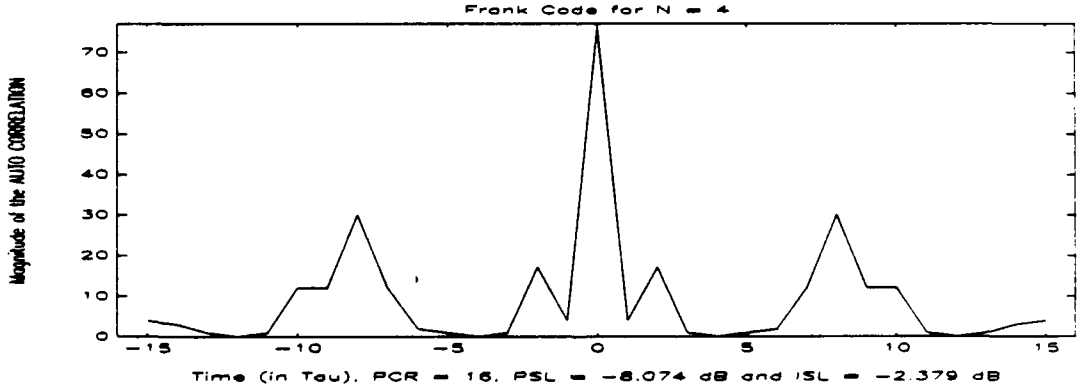


Figure 52: Output of the correlator for a Frank code of n=4, (thus N=16) without doppler effects

uncorrected doppler or with corrected doppler. There is an error message (ERR2MSG) which detects improper data entries, notifies the operator and restarts the process. Then the choice of modulation scheme is made, and this can be seen in Figure 54, which is an expanded portion of Figure 53. Due to the peculiarities of MATLAB, it is necessary to have three separate sub programs for each modulation scheme to account for the presence of no doppler, uncorrected doppler or corrected doppler. It is possible to exit the simulation from either the main or sub programs. This structure was used since it allowed the best user interface and is a robust format, that is data entry error tolerant. The other advantage of this format is that should an operator desire to change one of the sub programs or add another sub program the system can accommodate this. Any errors made in this process will affect only the subprogram in question. The rest of the simulation will remain operational while the new addition is being debugged.

C. DOPPLER CORRECTION SCHEME FOR PULSE COMPRESSION

When the transmitted radar pulse encounters a moving target, a doppler shift occurs in the reflected pulse. The target's relative velocity to the radar causes a shift in frequency given by:

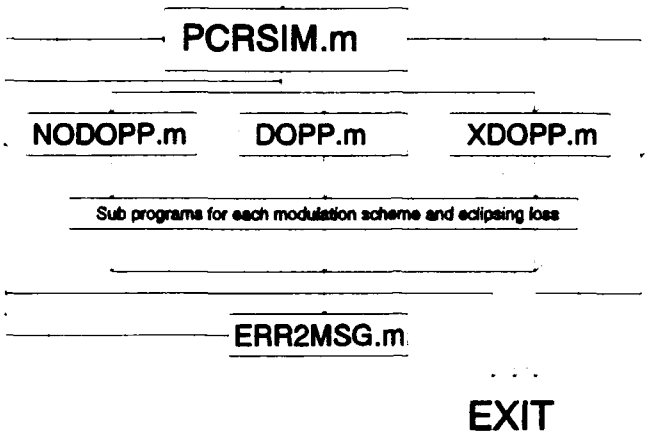


Figure 53 : Overall program structure for PCR simulation

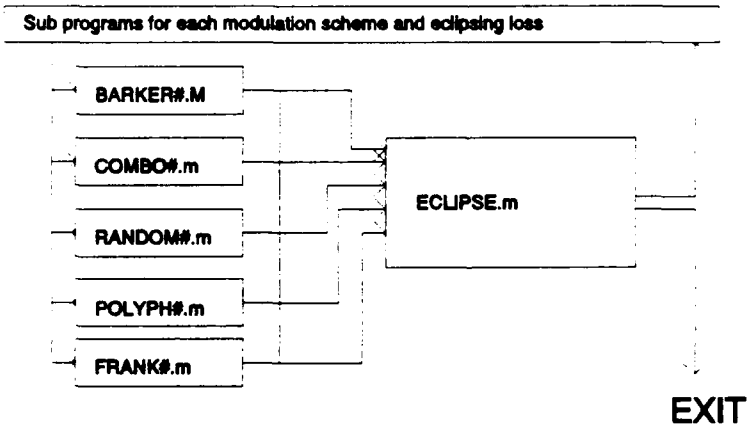


Figure 54 : Sub program details of PCR simulation

$$f_d = \frac{2\ V_d}{\lambda}. \qquad (56)$$

The corresponding change in phase is given by:

$$\phi_d = 2\pi f_d\, n\tau \quad for\ n=1\text{-}N,\ n \in I \qquad (57)$$

where

- V_d = the relative doppler velocity between the target and the radar,
- f_d = the doppler frequency resulting from V_d,
- λ = the wavelength of the transmitted pulse,
- ϕ_d = the doppler phase shift resulting from V_d,
- n = the subpulse number, and
- τ = the subpulse width.

If the radar platform is itself in motion this cause an additional doppler shift as well. However, since the speed of the radar platform is known, it can be readily countered in the RSP calculations and is therefore not included in the following equations. One of the simplest schemes in common use is to attempt to counter the effect of the doppler shift by introducing another shift of equal magnitude, but opposite in sign. Thus if a doppler shift is negative, a positive shift of the same magnitude is used to cancel it out.

This scheme is demonstrated in the XDOPP.m subprogram option of the PCRSIM simulation program. The concept behind this correction scheme is as follows;

- If the radar itself is in motion this speed will be known, and it can be accounted for in the assumed counter doppler equation.
- The highest possible speed (V_{MAX}) of the target to be detected is known or can be assumed. One half of this speed is then used in the counter doppler equation at Equation 74 for V_c. The assumption of one half maximum is made since in most situations this will approximate the likely speeds encountered.
- A decision as to whether the radar will be encountering opening or closing targets is critical. If the wrong decision is made then the overall doppler shift encountered will be worse than if no correction were made. If the target's doppler shift is unambiguous to the radar's filter bank the correct decision will be made. However, if the dop pler frequency is ambiguous than it is quite possible that the wrong decision will be made.
- If the correct decision as to an opening or closing target has been made this simple technique is surprisingly effective in removing the worst effects of doppler.

The equations for this correction scheme are as follows:

$$V_c = \frac{1}{2} V_{MAX} , \qquad (58)$$

which results in the overall doppler shift the waveform sees being (if the correct decision as to an opening or closing target is made)

$$f_d = \frac{2\ (V_d - V_c)}{\lambda} \qquad (59)$$

or (if the wrong decision is made)

$$f_d = \frac{2\ (V_d + V_c)}{\lambda}. \qquad (60)$$

The corresponding change in phase uses the new f_d from Equation 59 or 60 and is given by:

$$\phi_d = 2\pi f_d\, n\tau \quad for\ n=1 \to N,\ n \in I, \qquad (61)$$

where V_c is the counter velocity and V_{MAX} is the maximum velocity of the target.

D. OPERATING INSTRUCTIONS AND GUIDE

1. Initial Setup

To use this simulation, the following MATLAB.m files must be copied into the user sub directory. Copies of these programs are available from Dr GS Gill at USNPGS.

- PCRSIM.m is the simulation control program.
- NODOPP.m is the simulation subprogram responsible for processing a modulation scheme without the presence of doppler.
- DOPP.m is the simulation subprogram responsible for processing a modulation scheme with doppler.
- XDOPP.m is the simulation subprogram responsible for processing a modulation scheme with corrected doppler.
- BARKER1.m is the simulation subprogram responsible for processing the Barker modulation schemes without the presence of doppler.

- BARKER2.m is the simulation subprogram responsible for processing the Barker modulation schemes with doppler.
- BARKER3.m is the simulation subprogram responsible for processing the Barker modulation schemes with corrected doppler.
- COMBO1.m is the simulation subprogram responsible for processing the combined Barker modulation schemes without the presence of doppler.
- COMBO2.m is the simulation subprogram responsible for processing the combined Barker modulation schemes with doppler.
- COMBO3.m is the simulation subprogram responsible for processing the combined Barker modulation schemes with corrected doppler.
- RANDOM1.m is the simulation subprogram responsible for processing the pseudo random modulation schemes without the presence of doppler.
- RANDOM2.m is the simulation subprogram responsible for processing the pseudo random modulation schemes with doppler.
- RANDOM3.m is the simulation subprogram responsible for processing the pseudo random modulation schemes with corrected doppler.
- POLYPH1.m is the simulation subprogram responsible for processing the P3 modulation schemes without the presence of doppler.
- POLYPH2.m is the simulation subprogram responsible for processing the P3 modulation schemes with doppler.
- POLYPH3.m is the simulation subprogram responsible for processing the P3 modulation schemes with corrected doppler.
- FRANK1.m is the simulation subprogram responsible for processing the Barker modulation schemes without the presence of doppler.
- FRANK2.m is the simulation subprogram responsible for processing the Barker modulation schemes with doppler.

- FRANK3.m is the simulation subprogram responsible for processing the Barker modulation schemes with corrected doppler.

- ECLIPS.m is the simulation subprogram responsible for displaying the eclipsing loss parameters after one the modulation subprograms has been calculated.

- ERR2MSG.m informs the operator when an entry error has been made and restarts DFTBANK after clearing the system of all variables.

This simulation will work on a DOS 286 machine, but a 386 or higher is preferable for increased speed. To start the simulation type PCRSIM [ENTER] at the MATLAB prompt.

2. Printing Graphical Outputs

The assumption has been made that the PRINT SCREEN key will be used to print out plots. To use this key the DOS graphics command must have been issued at the DOS prompt before any printing is attempted. C:\>> graphics [ENTER] is the necessary command.

Should it be preferable to create meta files for each plot the operator has two options. He can either

- Modify all plot options in the subprograms by adding a meta command and file name after the plot command, or

- Press the Ctrl-C keys to exit the simulation once the plot is on the screen. This will return him to the MATLAB prompt and he can issue the meta command from there. To restart the simulation type PCRSIM [ENTER].

3. Program Options

After the PCRSIM command has been issued to start the simulation, the first screen seen by the operator is at

Figure 55. The second screen the operator will see is at Figure 56. At this point the operator must choose which

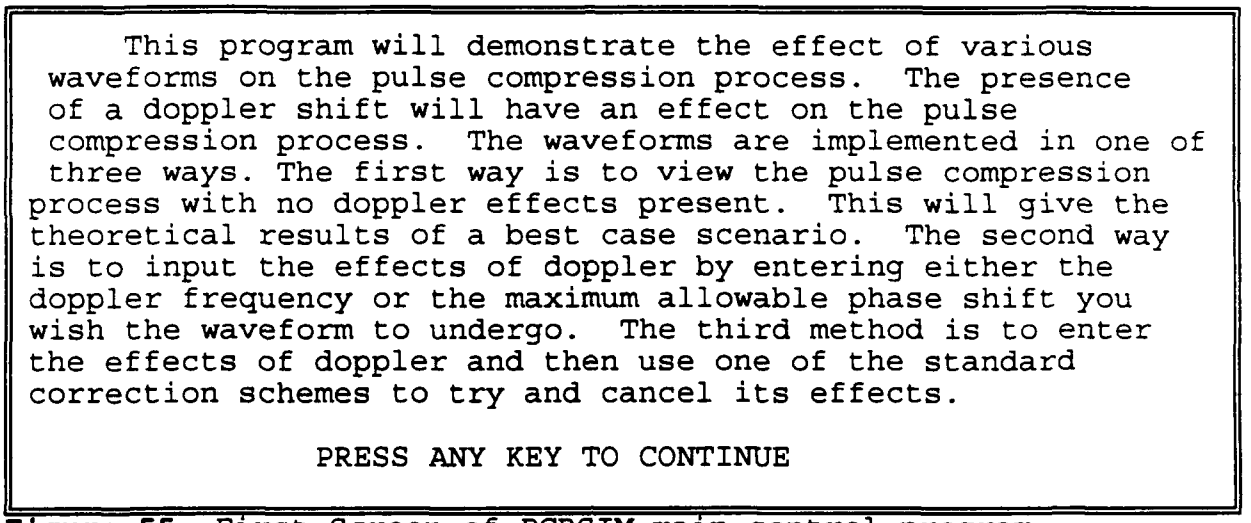


Figure 55: First Screen of PCRSIM main control program

type of doppler processing the modulation scheme is to undergo.

Once this has been done the main program transfers control to the subprogram selected in Figure 56. All of the screens for the subprograms are very similar, the only difference being what are the required inputs.

The screen at Figure 57 shows the selection choice for the available modulation schemes if the no doppler choice was selected in Figure 56. The additional inputs required for the doppler or corrected doppler subprograms will be discussed further on. Once one of the modulation schemes has been selected than the required inputs will be

```
 Which doppler technique do you wish to use?

To see pulse compression without doppler,          ENTER 1
To see pulse compression with doppler,             ENTER 2
To see pulse compression with corrected doppler,ENTER 3
 To exit the simulation,                           ENTER 0
              PRESS ANY KEY TO CONTINUE
```

Figure 56: Second Screen of PCRSIM main control program

```
 You have chosen not to consider doppler sensitivity.
 You now need to choose the waveform you wish to process.
      For a SINGLE BARKER CODE, choose "n=1", or
      For a COMBINED "m in n" BARKER CODE, choose "n=2", or
      For a Pseudo Random code, choose "n=3", or
      For a Polyphase code, choose "n=4", or
  For a Frank code, choose "n=5", or
  To return to the main menu, choose "n=6", or
      To quit, choose "n=0"
         PRESS ANY KEY TO CONTINUE
ENTER the # of your waveform choice, n =
```

Figure 57 : The no doppler modulation scheme screen

requested for that code as discussed in Section A of this chapter. The screen for a Barker code is at Figure 58.

```
The known Barker codes have lengths of n= 2,3,4,5,7,11 or
13.  You need to choose the length of the Single Barker code
you want.

Then press RETURN

ENTER the # of the Single Barker Code to use, n =
```

Figure 58 : The single Barker modulation scheme screen

The results will be as for Figure 48 if a single Barker code of length 13 is selected. The next option is to use the eclipsing loss subprogram from the choices presented at Figure 59.

```
If you want to do a Single Barker Code again, choose n = 1
 If you want to return to the non-doppler menu, choose n = 2
 If you want to return to the Main menu, choose n = 3
 If you want to view the PRI and pulse length, choose n = 4
 If you want to quit, choose n = 0
 Then press RETURN
 Choose n =
```

Figure 59 : Final screen of a modulation scheme subprogram

If the ECLIPS.m subprogram has been chosen then the screen at Figure 26 will request the necessary inputs. An example of the output of this subprogram is at Figure 27 for RF =25 MHz, PRF = 5 KHz and τ = 5 cycles of the RF for a PCR = 169. This plot shows the pulse width as compared to the

```
    You will need to input values for the RF frequency
  of operation of the system, the PRF you wish to consider
  and the # of cycles of the RF that one Tau is to be
  comprised of.  The PCR will either be given by a higher
  level program or input by you.


 Enter the RF freq of operation in [ MHz ], RF =
 Enter the PRF in [ KHz ], PRF =
 Enter the # of cycles of RF that is 1 Tau, # =
 Enter 1 if PCR present or 2 to input it, choice =
```

Figure 60 : First screen of the ECLIPS.m subprogram

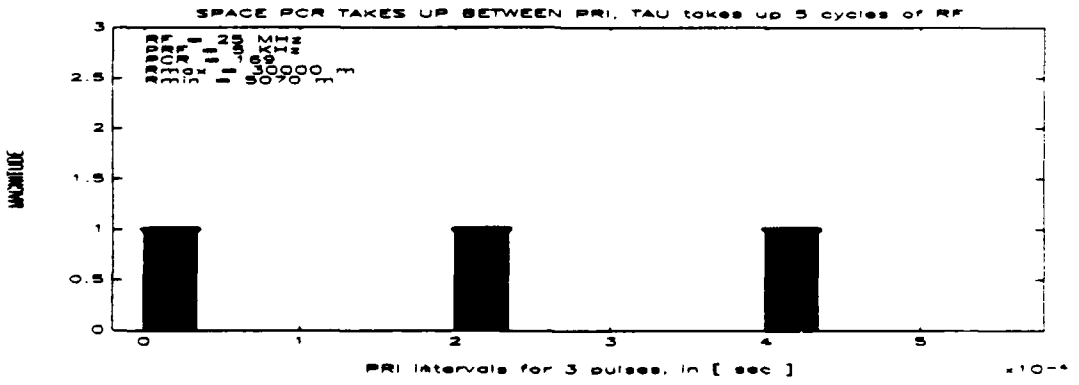


Figure 61 : Eclipsing loss graphical presentation over 3 PRI

PRI for three pulses. The unambiguous range and the minimum range are calculated and displayed for these inputs. After this the operator has the option to return to the main menu or to redo the eclipsing loss option.

Should any data entry error have occurred that the inherent MATLAB error routines did not recognize, then the simulation will detect it and the error message at Figure 62 will be displayed. All variables will be cleared, and the main program will be restarted.

```
YOUR CHOICE DOES NOT CORRESPOND TO THOSE AVAILABLE.
THE PROGRAM WILL BE RESTARTED FOR YOU.
REMEMBER YOU MUST CHOOSE 0 TO EXIT.


PRESS ENTER TO CONTINUE...
```

Figure 62: Error message subprogram screen

The additional input screens necessary for the doppler pulse compression options are at Figures 63, 64 and 65. These screens appear after the modulation choice has been made because they need to use some of the inputs from the modulation scheme. The first screen of the doppler sensitivity choices available is at Figure 63. The choice is between entering the doppler frequency for a specific case or entering the overall phase shift to test a code's response to any system that could have a doppler velocity which would produce such a phase shift.

The additional input screens necessary for the corrected doppler pulse compression options are at Figures

```
You must now decide whether you want to see the
result for a doppler frequency, or for a maximum phase
shift across the coding sequence.

By giving the maximum phase shift, you will be
controlling the amount of error in the processed signal.
With this option, your maximum doppler will be dictated.

By giving a doppler frequency, you will see the error
caused by that doppler.

Enter 1 to specify doppler shift, or 2 for phase error.
```

Figure 63 : Doppler choice screen of the DOPP.m subprogram

```
Enter doppler freq. in HZ

Enter the compressed pulse width tau in microsec.
```

Figure 64 : Doppler choice screen if doppler shift selected

```
Enter the maximum allowable phase shift in radians
```

Figure 65 : Doppler choice screen if phase error selected

66 and 67. For the first screen at Figure 66, the point to note is that a positive doppler velocity entry means that

the correct decision as to whether it is an opening or closing target was made. A negative entry means that an opening target was identified as being a closing target or vice versa, and thus the consequences of an error will be seen.

```
    You have chosen to use the Doppler sensitivity
feature. The simulation will assume that the correct
decision as to whether it is an opening or closing target
has been made if the doppler frequency entered is
positive.  If the doppler frequency entered is negative
then you will see the consequences of having made
the wrong decision.

    You must now enter the doppler data the program will
require.
```

Figure 66 : Doppler choice screen if XDOPP.m selected

```
 Enter target maximum speed in knots
 Enter actual target speed in knots
 Enter the RF operating frequency in MHZ
 Enter the PRF in KHZ
 Enter the compressed pulse width in microseconds

     Compensation for the doppler shift by "derotating"
using an assumed derotation doppler of one-half target
maximum speed will be done.
```

Figure 67 : Data entry screen if XDOPP.m selected

E. INTERPRETATION OF RESULTS

The output of the simulation is mostly graphical in nature since graphical displays are readily assimilated and provide a more intuitive understanding of the process. Results from each of the subprograms will be presented.

These plots will be analyzed to demonstrate some of the capabilities of the simulation program and to provide worked examples as a reference guide.

1. Pulse Compression Without Doppler

The subprograms without doppler are all quite self explanatory, and examples for each modulation scheme are at Figures 48, 49, 50, 51 and 52. This selection allows the operator to select a coding scheme that meets his ideal needs, and the other two options allow him to see how this code performs under doppler conditions and if a correction scheme can help its performance.

2. Pulse Compression With Doppler

The subprograms with doppler allow that either a specific case (doppler frequency specified) or a general case (maximum allowable phase shift specified) be investigated. A combined 13 in 13 Barker that has undergone a maximum phase shift of 90° is at Figure 68. The output of the correlator has degenerated so much as compared to the ideal case at Figure 49 that it is useless for detection purposes. A P3 polyphase scheme with the same PCR as that of the combined Barker is at Figure 69. Here the output of the correlator is very similar to that of the ideal case at Figure 51. Note however that a range error has been introduced as the mainlobe of the response of the correlator

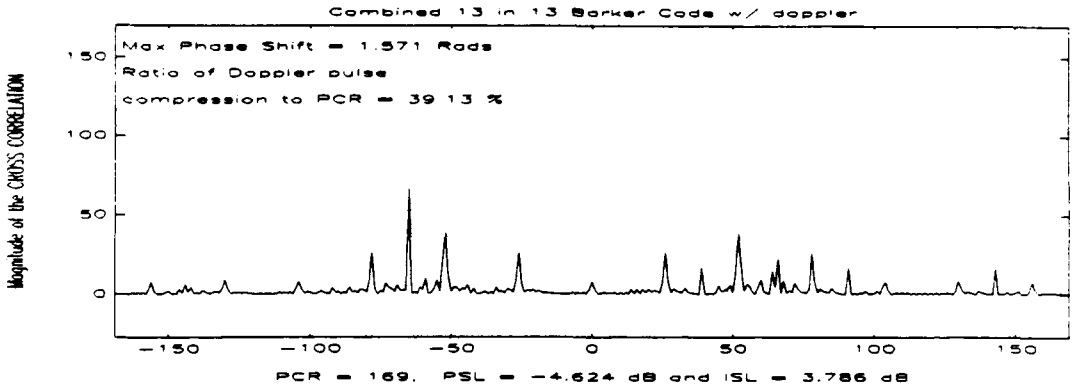


Figure 68 : 13 in 13 Combined Barker with 90° of phase shift overall

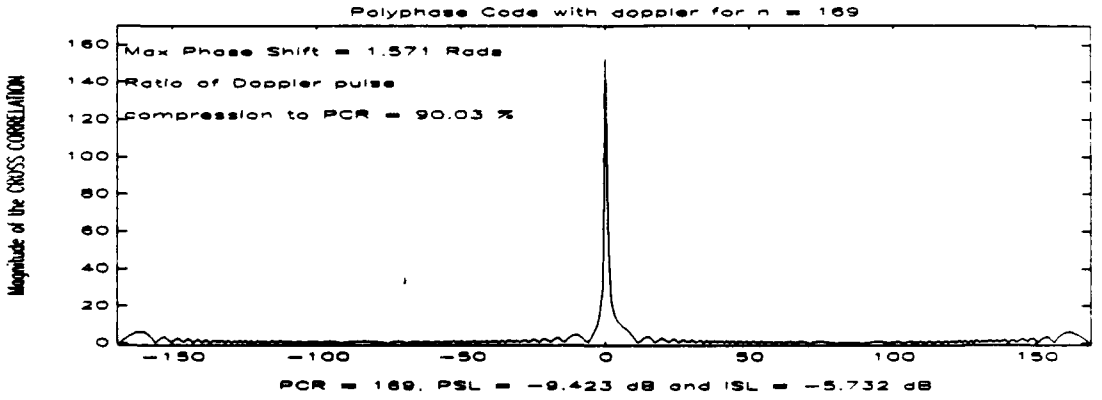


Figure 69 : 169 component P3 polyphase code with 90° of phase shift overall

has been shifted in time. If the doppler shift that caused this is known, then the range error can be corrected.

3. Pulse Compression With Corrected Doppler

The subprograms with corrected doppler are always done for a specific case. A combined 13 in 13 Barker that has undergone a doppler phase shift is used in the correction scheme with the following parameters:

- RF=2 GHz
- PRF= 13 KHz
- Tau = 1 μSec
- V_{tgt} = 700 KPH
- V_{max} = 900 KPH.

The output is at Figure 70, and clearly this simple doppler correction scheme was very effective. The output at Figure 71 shows what the result would have been without the doppler correction being applied. This was done simply by letting V_{max} = 0 which gives the report for uncorrected doppler. The response at Figure 70 is a useable compression, albeit somewhat less than ideal. The correlator output at Figure 71 would likely result in two targets at different ranges being declared rather than the one target that is actually there. The correlator output at Figure 72 is what would result if a wrong decision as to an opening or closing target had been made. This was done by entering f_{tgt} = -700 KPH.

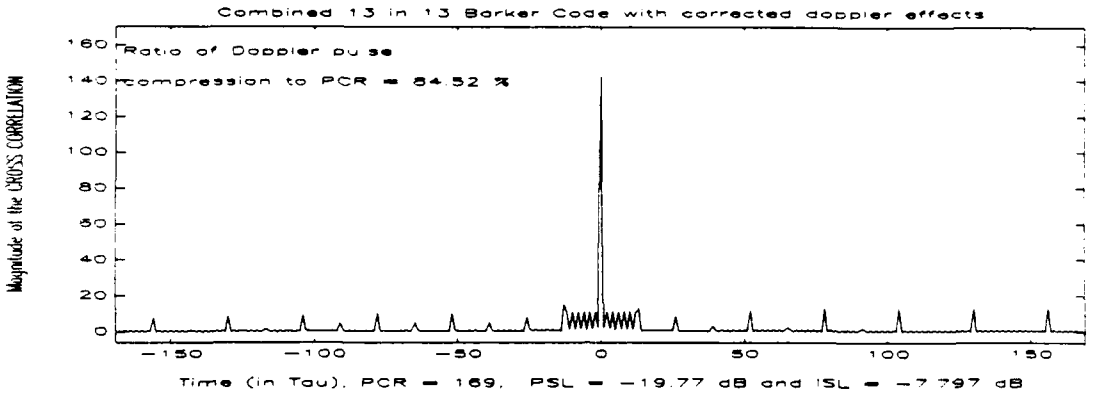


Figure 70 : Corrected doppler after pulse compression for a 13 in 13 combined Barker code. The pulse compression achieved is 85% of the PCR.

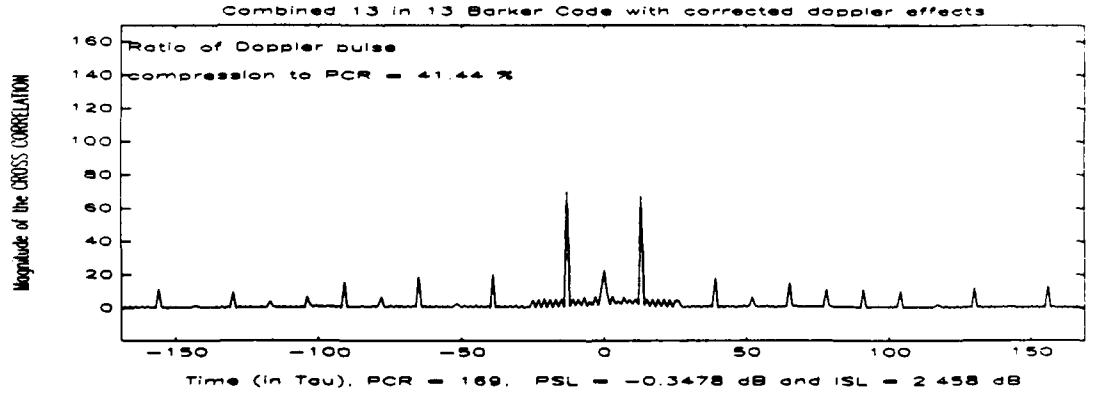


Figure 71 : Uncorrected doppler after pulse compression for a 13 in 13 combined Barker code. The pulse compression achieved is 41% of the PCR.

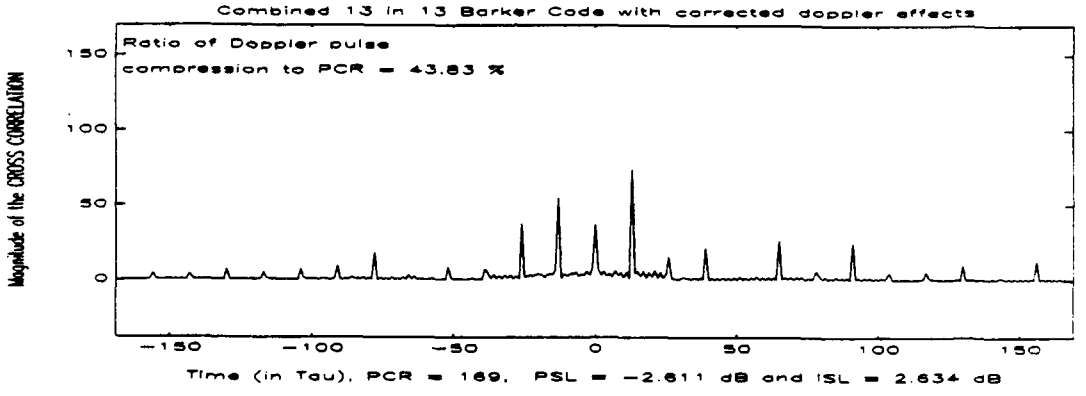


Figure 72 : Corrected doppler after pulse compression for a 13 in 13 combined Barker code where an incorrect decision as to whether it is an opening or closing target was made. The pulse compression achieved is 43% of the PCR.

VIII. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

This thesis has shown the background to the development of the DFT as a filter bank for radar applications. This development was then simulated in MATLAB for single filters, filter banks and the generation of target reports. In order to reduce the sidelobe response of the filters, weighted inputs through the use of windows were developed and simulated for a weighted DFT filter bank. Finally, the concept of pulse compression was addressed and various representative pulse compression schemes were simulated under a variety of doppler conditions.

Traditional classroom methods are not sufficient to demonstrate radar signal processing beyond trivial cases. The use of simulation allows the student the freedom to go beyond the trivial cases and arrive at a fuller understanding of the principles involved. Simulation on a personal computer is simple, cost effective and accessible.

These simulation programs are freely available for student use and with them the computational burden of addressing some of the real world issues of radar signal processing has been removed. The simulations deliver the same degree of accuracy as the real RSP processes would,

albeit at a much slower pace than the on board computers of a modern radar. The simulation programs are available from:

Professor G.S. Gill,

Code EC/Gl

Naval Postgraduate school,

Monterey, CA 93943

B. RECOMMENDATIONS

Some of the possibilities available for further thesis study in this area can be seen from the algorithm processing blocks in Figure 2. In particular, the simulation of the processing blocks necessary for the adaptive nulling, CFAR detector and ambiguity resolution processes are considered to be a logical extension to this thesis.

LIST OF REFERENCES

1. Stimson, GW, *Introduction to Airborne Radar*, Hughes Aircraft Company, El Segundo, CA, 1983

2. Lewis, BL, Kretschmer Jr, FF and Shelton, WW, *Aspects of Radar Signal Processing*, Artech House, Norwood, MA, 1986

3. Skolnik, M., *Radar Handbook*, McGraw Hill, 1990

4. Morris, GV, *Airborne Pulsed Doppler Radar*, Artech House, Norwood, MA, 1988

5. Burton, DK, Cook, CE and Hamilton, P., *Radar Evaluation Handbook*, Artech House, Norwood, MA, 1991

6. Skolnik, M., *Introduction to Radar Systems*, McGraw Hill, 1980

7. Strum, RD and Kirk, DE, *Discrete Systems and Digital Signal Processing*, Addison-Wesley Publishing Company, New York, NY, 1989

8. Reid, JG, *Linear System Fundamentals*, McGraw Hill, 1983

9. Haddad, RA and Parsons, TW, *Digital Siganl Processing*, W.H. Freeman and Company, New York, NY, 1991

10. Chan, HC, *Implementation of High Speed Fast Fourier Transforms (FFT) for Radar Signal Processing*, Technical Report CRC-1394, Communications Research Centre, Ottawa, Ont., August 1985

11. Behroozi, V., *Implementation of FFT and Pulse Compression Routines on the SPT Frequency Domain Array Processor*, Technical Report DREO-1041, Defense Research Establishment, Ottawa, Ont., September 1990

12. Barcia, JPG, *Survey on Modern Radar Signal Processing*, Thesis, Naval Postgraduate School, Monterey, CA, December 1975

13. Evans, JE, *Storm Models for End to End TDWR (Terminal Doppler Weather Radar) Signal Processing Simulation Tests*, Report No ATC-155, Massachusetts Institute of Technolgy, May 1989

14. Couch II, LW, *Digital and Analog Communication Systems*, MacMillan Publishing Company, New York, NY, 1987

INITIAL DISTRIBUTION LIST

1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 52 Naval Postgraduate School Monterey, CA 93943-5000	2
3.	Chairman, Code EC Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5000	1
4.	Professor G.S. Gill, Code EC/Gl Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5000	3
5.	Professor M. Tummala, Code EC/Tu Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5000	1
6.	DLAEEM 4-3 National Defense Headquarters Ottawa, Ontario, Canada K1A 0K2	2
7.	SO ATV Canadian Defense Liason Staff (Washington) 501 Pennsylvania Ave NW Washington, DC 20001	2
8.	Major Paul A Ohrt 1681 Tache Way Orleans, Ontario, Canada K4A 2T7	1